e*Xchange eBusiness Integration Suite

Onyx/SAP Intelligent Bridge

*e*Gate 4.1*

The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, STC assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of STC; and may be used only in accordance with the License Agreement, as protected and enforceable by law. STC assumes no responsibility for the use or reliability of its software on equipment that is not supported by STC.

Contents

Chapter 1

Introduction

1.1 About this Manual

1.1.1 Overview

The Onyx/SAP Intelligent Bridge is a software solution that creates a communication link between SAP and Onyx database applications, enabling the two databases to exchange data. This manual provides a functional description of the Onyx/SAP Intelligent Bridge, as well as installation and configuration instructions.

To get the Onyx/SAP Intelligent Bridge operating, follow the instructions provided in **"Installation and Configuration" on page 9**. To gain a technical understanding of how the Onyx/SAP Intelligent Bridge operates, read the chapters relating to the module(s) about which you want to learn.

1.1.2 Intended Reader

This manual is intended for developers or system administrators with expert-level knowledge of the following:

- e*Gate
- Windows NT operations and administration
- Onyx Front Office applications
- SAP R/3 enterprise applications

1.1.3 Content Summary

This manual is divided into the following chapters:

- **Introduction** on page 1 - An overview of the general functionality and components of the Onyx/SAP Intelligent Bridge.

- **Installation and Configuration** on page 9 - Instructions for installing and configuring the Onyx/SAP Intelligent Bridge and its associated components.
- **Module 1: The Product Bridge** on page 22 — A functional description of the Product Bridge module, intended for developers who want to customize this module.
- **Modules 2 and 3: The Customer Bridges** on page 46 — A functional description of the Customer Bridge modules, intended for developers who want to customize these modules.

1.1.4 Supporting Documents

The following documents contain information related to this document:

- *e*Gate System Administration and Operations Guide*
- *e*Gate Installation Guide*
- *SAP ALE e*Way User's Guide*
- *SAP/BDC e*Way User's Guide*
- *DART e*Way for ODBC User's Guide*
- *SAP Integration Guide*

1.2 Executive Summary

1.2.1 Background

Different divisions of companies often use various types of software to accomplish separate business goals. SAP applications are widely used by production oriented divisions, while Onyx applications are commonly used by sales divisions within the same company. Even though the fundamental goals of these divisions differ, there is still a need to share many types of data to accomplish the main goal of making the company profitable. Therefore, employees using SAP need access to data that are only available in Onyx, and vice versa.

The basic structures of the SAP and Onyx databases are similar; both applications store data in tables within their own databases. They cannot share data as stand-alone applications, however, because the methods each application uses to handle the data differ. In order to exchange data with each other, the SAP and Onyx applications must be integrated with the Onyx/SAP Intelligent Bridge.

1.2.2 The Onyx/SAP Intelligent Bridge

The Onyx/SAP Intelligent Bridge is a software solution that creates a communication link between the SAP and Onyx database applications. This communication link enables the automatic and seamless transfer of data from specific fields in the SAP database to specific fields in the SAP database. Additionally, new Customer records created in the Onyx application are transferred to the SAP database via this communication link. Once the bridge is installed and configured, changes to data in the SAP database are implemented in the Onyx database, and new Customer records created in the Onyx database appear in the SAP database.

Functional Overview

The transfer of data between the SAP and Onyx databases is a three step process executed by the Onyx/SAP Intelligent Bridge. First, data is pulled from database A. The data is then restructured into a format that is compatible with database B. Finally, the data is placed into database B, completing the data transfer.

The SAP database exports and imports data in a format called Intermediate Document (IDoc). The Onyx/SAP Intelligent Bridge extracts data from the SAP database in IDoc format and restructures the data into Outside Database Connectivity (ODBC) format, which is recognized by Onyx. The data is then placed into Onyx database tables.

The Onyx database exports and imports data in ODBC format. The Onyx/SAP Intelligent Bridge pulls data from the Onyx database in ODBC format, and translates it into IDoc format. The data is then placed into SAP database tables.

Components

When you install the Onyx/SAP Intelligent Bridge, you are actually installing several software components that form a "bridge" between the SAP and Onyx database applications. These components include e*Gate, IQs, schemas, and specific types of e*Ways e*Gate needs to communicate with the SAP and Onyx database applications.

e*Gate

e*Gate is GUI-driven, Enterprise Application Integration (EAI) software for integrating client/server/web technologies. It serves as the integration backbone of the Onyx/SAP Intelligent Bridge. e*Gate configures and connects software components called e*Ways, Intelligent Queues (IQs), and Business Object Brokers (BOBs) together to form schemas. Once a schema is created, e*Gate can integrate various applications.

e*Gate components exchange data using a publication/subscription method. This means that after an e*Gate component receives and processes data, it transfers the data to other components to which it is configured to publish the data. e*Gate components receive data from other components to which they are configured to subscribe.

For information on e*Gate, see the *e*Gate System Administration and Operations Guide.*

e*Ways

e*Ways establish connectivity with external applications by receiving, restructuring, and sending data to external systems, IQs, or BOBs. e*Ways can either be configured to be inbound or outbound. If an e*Way is configured as inbound, it extracts data from an external system and passes the extracted data to an IQ or BOB. If an e*Way is configured as outbound, it receives data from an IQ or BOB and places the data into an external system.

e*Ways use Collaborations. Collaborations are instructions that tell the e*Way how to translate and restructure data, and where to route the data once it is processed. e*Gate is used to configure e*Way Collaborations.

The Onyx/SAP Intelligent Bridge software solution uses the following e*Ways:

- SAP ALE e*Way — Enables e*Gate to exchange data with the SAP database application using Application Link Enabling (ALE). For information on the SAP ALE e*Way and Application Link Enabling, see the *SAP ALE e*Way User's Guide.*
- SAP BDC e*Way — Enables e*Gate to exchange data with the SAP database application using the Batch Data Communication (BDC) method. For information on the SAP BDC e*Way and Batch Data Communication, see the *SAP BDC e*Way User's Guide.*
- ODBC e*Way — Enables e*Gate to exchange data with the Onyx database application using the Outside Database Connectivity (ODBC) method. For information on the ODBC e*Way and Outside Database Connectivity, see the *ODBC e*Way User's Guide.*

Intelligent Queues (IQs)

IQs manage the exchange of information between components within the e*Gate system. They receive data that inbound e*Ways extract from external applications, and route the data to outbound e*Ways. They can also exchange data with BOBs. For more information on IQs, see the *e*Gate System Administration and Operations Guide.*

Business Object Brokers (BOBs)

Like e*Ways, BOBs use Collaborations to route, translate, and restructure data within the e*Gate system. BOBs can only exchange data with e*Ways and IQs, however, while e*Ways can exchange data with external systems.

Schemas

Schemas are files and associated stores created by e*Gate that contain the parameters of all the components that control, route, translate, and restructure data as it moves through e*Gate. After you install the Onyx/SAP Intelligent Bridge software, you must use e*Gate to either implement one or more of the included schemas, or create a new schema. If you choose to create a new schema, use e*Gate to configure the e*Ways and IQs, and establish a connection between the SAP and Onyx database applications.

For more information on schemas, see the *e*Gate System Administration and Operations Guide.*

Once you have implemented a schema, the Onyx/SAP Intelligent Bridge is complete; changes (creates, edits, and deletes) to data in specific SAP database tables appear in corresponding Onyx database tables. Also, new Customer records created in Onyx appear in corresponding SAP database tables.

Architecture

Modules

The Onyx/SAP Intelligent Bridge is a single "bridge" divided into three modules: one module that handles the transfer of Material/Product related data, and two modules that handle the transfer of Customer/Accounts related data. These modules are:

- SAP to Onyx Product Bridge — Enables data transfers of Material related data from the SAP database to tables containing Product related data in the Onyx database.
- SAP to Onyx Customer Bridge — Enables data transfers of Customer related data from the SAP database to tables containing Company related data in the Onyx database.
- Onyx to SAP Customer Bridge — Enables the transfer of new Customer records from the Onyx database to tables in the SAP database.

Depending on your needs, you can install all of these modules at one time, or you can install the just the Product Bridge module or just the Customer Bridge modules. If you are going to install either of the Customer Bridge modules, however, you must install the other Customer Bridge module as well.

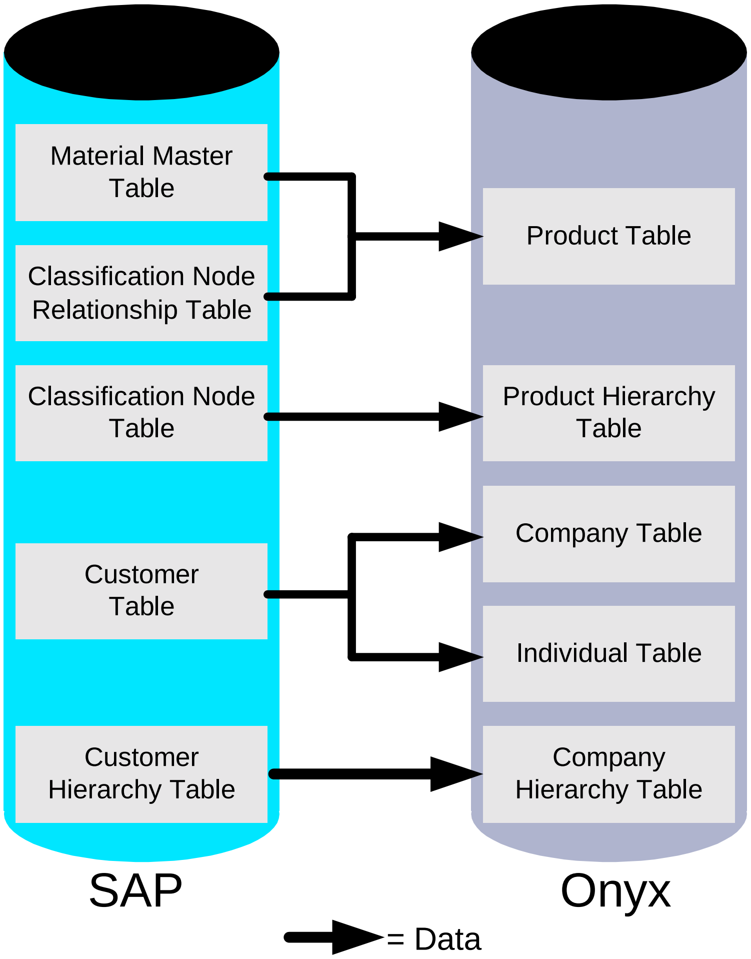
Data Paths

Each module contains one or more data paths, which are used to transfer data. Data paths are groups of e*Ways, IQs, and BOBs working together to create communication links between corresponding tables existing in separate databases. You create data paths by using e*Gate to configure the components of the Onyx/SAP Intelligent Bridge to form communication links between corresponding tables in the SAP and Onyx databases.

A database table is considered to correspond to another table if both tables contain similar data. While many tables in the SAP and Onyx database correspond to each other, some do not. Therefore, to avoid confusion, the Onyx/SAP Intelligent Bridge only enables you to create data paths between SAP and Onyx tables that correspond to each other.

Figure 1 on page 6 illustrates the SAP tables and their corresponding Onyx tables, as well as the data paths that connect them.

Figure 1 SAP to Onyx Data Paths



In addition to transferring data from SAP tables to Onyx tables, the Onyx/SAP Intelligent Bridge also uses data paths to transfer new Customer records from the Onyx database to the SAP database. These new Customer records are created in the SAP Customer Out table in the Onyx database and transferred to the SAP database. When the SAP database receives the new Customer record data, the IDoc automatically distributes the data to the appropriate SAP tables.

Figure 2 on page 7 illustrates the transfer of new Customer records from the Onyx database to the SAP database.

Figure 2 Onyx to SAP Data Path

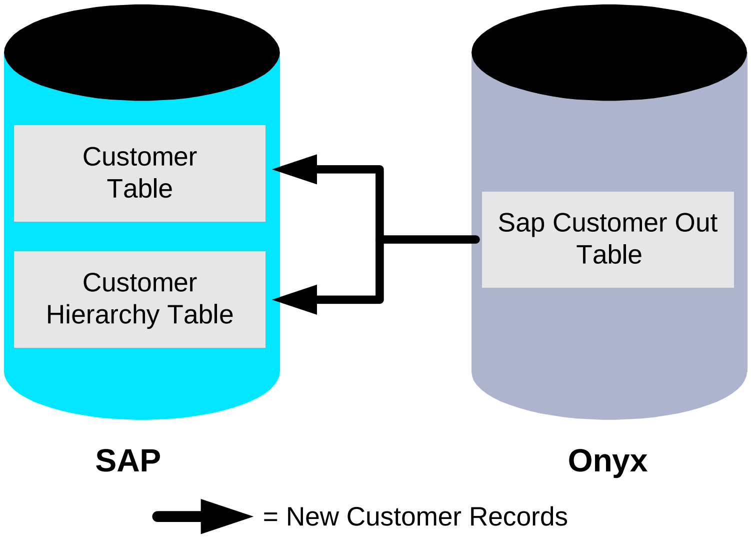


Table 1 describes which data paths are included with each module.

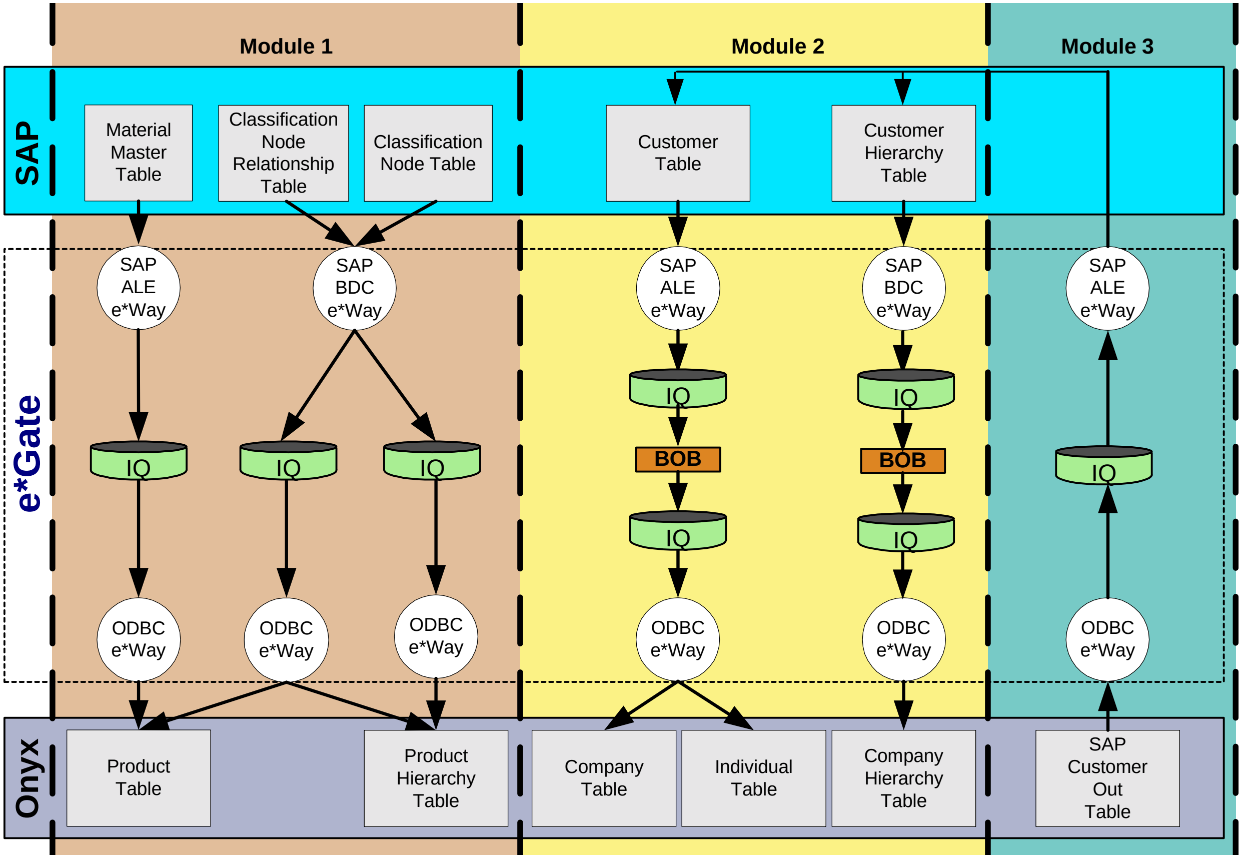
Table 1 Data Paths

Module	Data Paths
SAP to Onyx Product Bridge	▪ Material Master Table to Product Table ▪ Classification Node Relationship Table to Product Table ▪ Classification Node Table to Product Hierarchy Table
SAP to Onyx Customer Bridge	▪ Customer Table to Company Table ▪ Customer Table to Individual Table ▪ Customer Hierarchy Table to Company Hierarchy Table
Onyx to SAP Customer Bridge	▪ SAPCustomerOut to the Customer Table and Customer Hierarchy Table.

Data paths are uni-directional, meaning data can only be transferred in one direction along each path. All data paths in the Onyx/SAP Intelligent Bridge transfer data from the SAP database to the Onyx database, with the exception of the data path in the Onyx to SAP Customer Bridge module. Therefore, you must edit and delete all data only in the SAP database. You must also create all data exclusively in the SAP database, except for new Customer records, which must be created in the Onyx database. If the Onyx application is used to make any changes to shared data other than the creating of new Customer records, the SAP and Onyx databases become asynchronized, defeating the purpose of the Intelligent Bridge.

Figure 3 on page 8 illustrates the architecture of a complete Onyx/SAP Intelligent Bridge.

Figure 3 The Onyx/SAP Intelligent Bridge Architecture



Chapter 2

Installation and Configuration

2.1 Pre-Installation

Administrator Privileges

Verify that you have Administrator privileges for the computer on which you plan to install e*Gate and the Onyx/SAP Intelligent Bridge. If you are not sure, click the **Start** button and point to **Programs**->**Administrative Tools**->**User Manager**. Click on **Administrators** under the **Groups** section. If your user name is not listed, consult your system administrator.

System Requirements

Verify that the system on which you plan to install the Onyx/SAP Intelligent Bridge either meets or exceeds the minimum system requirements for e*Gate. For information on system requirements for e*Gate, see the *e*Gate Installation Guide*.

Environment Requirements

Verify that the environment in which you plan to install the Onyx/SAP Intelligent Bridge contains the following:

- SAP R/3 4.0 database application
- Onyx Customer Center 5.0

Bulk Load

Perform an initial bulk load of data from the SAP application to the Onyx application prior to installing the Onyx/SAP Intelligent Bridge. See the documentation provided with the SAP and Onyx software for instructions on how to perform a bulk load.

Microsoft SQL Server 7.0 Client Software

Install Microsoft SQL Server 7.0 Client software on the machine on which you plan to install (or have already installed) e*Gate, and setup a Data Source Name (DSN) for the Onyx database. See the SQL Server documentation for information on how to set up a Data Source Name.

2.2 Installation

If you do not have e*Gate installed on your system, follow the instructions in the Full Install section. If e*Gate is already installed on your system, follow the instructions in the Add-on Install section.

2.2.1 Full Install

1. Install e*Gate into an environment containing the following:
   - SAP R/3 4.0 database application
   - Onyx Customer Center 5.0

   For information on installing e*Gate, see the *e*Gate Installation Guide*.
2. Follow the on-screen instructions until you reach the **Select Components** window in the **e*Gate Add-on Applications** portion of the installation.
3. In the **Select Components** window of the **e*Gate Add-on Application** portion of the installation, check SAP to Onyx Product Bridge if you want to install the Product module. Check SAP to Onyx Customer Bridge and Onyx to SAP Customer Bridge if you want to install the Customer modules. Check all three if you want to install both the Customer and Product modules.

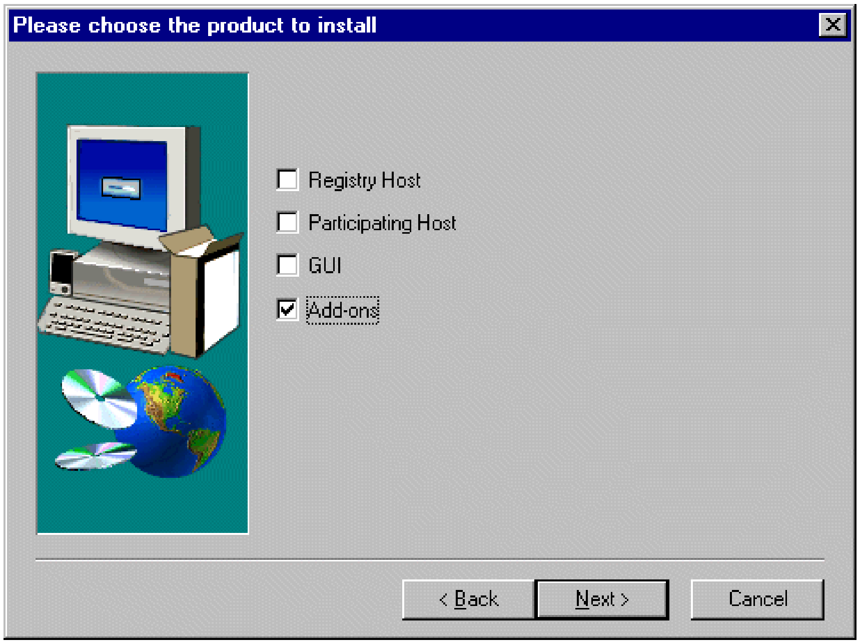
Note: *To enable the SAP and Onyx databases to exchange Customer data, you must install both Customer Bridge modules.*

4. Follow the on-screen instructions to complete the installation, then follow the procedures listed in **"Configuration" on page 11**.

2.2.2 Add-on Install

1. Insert the Onyx/SAP Intelligent Bridge CD and run setup. The on-screen instructions guide you through the **Welcome**, **Software License Agreement**, and **Information** windows.

2 When you reach the **Select Components** window, deselect all options except for **Add-ons**. The **e*Gate Add-on Application** portion of the e*Gate installation begins.



Please choose the product to install

□ Registry Host
□ Participating Host
□ GUI
☑ Add-ons

3 In the **Select Components** window of the **e*Gate Add-on Application** portion of the installation, check SAP to Onyx Product Bridge if you want to install the Product module. Check SAP to Onyx Customer Bridge and Onyx to SAP Customer Bridge if you want to install the Customer modules. Check all three if you want to install both the Customer and Product modules.

Note: *To enable the SAP and Onyx databases to exchange Customer data, you must install both Customer Bridge modules.*

4 Follow the on-screen instructions to complete the installation, then follow the procedures listed in **“Configuration” on page 11**

2.3 Configuration

After installing the Onyx/SAP Intelligent Bridge module(s), perform the following configuration procedures in the order they appear in this section.

2.3.1 ABAP Scripts

Follow the procedures below to transfer the ABAP Scripts to their appropriate destinations.

1 Copy K900470.s4x and R900470.s4x from the **sap_trans** directory (located in the e*Gate root directory) to the **/usr/sap/trans/bin** directory on the system hosting the SAP application.

2. Type **tp addtobuffer S4XK900470 <SID>** where SID is the SAP System ID of your SAP client.
3. Type **tp import S4XK900470 <SID> clientxxx u128** where **xxx** is the number of your SAP client number. (This command takes a few minutes to execute.) Ignore the **no profile used** message.

If you encounter errors during the import process, do the following:

- View the log files by going to the SAP application transaction **SE09**.
- Click the **Display Individually** button.
- Enter the transport number **M4XK900470**.
- Select **Goto->Transport Logs** from the pull-down menu.

2.3.2 Change Pointers

Activate change pointers in the SAP application to enable event-driven functionality.

- For the SAP to Onyx Product Bridge module, link the change pointers to the MATMAS03 IDoc structure.
- For the SAP to Onyx Customer Bridge module, link the change pointers to the DEBMAS03 IDoc structure.

See the documentation provided with your SAP application for information on activating change pointers.

2.3.3 Customer Hierarchy Functionality

(SAP to Onyx Customer Bridge only)

Standard SAP applications only allow a Sold-To Customer to have a Customer Hierarchy Node as its parent. Extend this functionality to allow a Hierarchy of all Sold-To Customer Master records, making it possible to assign a Sold-To record to another Sold-To record as the parent object. Use the following method to do this:

- Use transaction OVH2.
- Add a line for Account Group = 0001 and higher level Account Group = 0001.

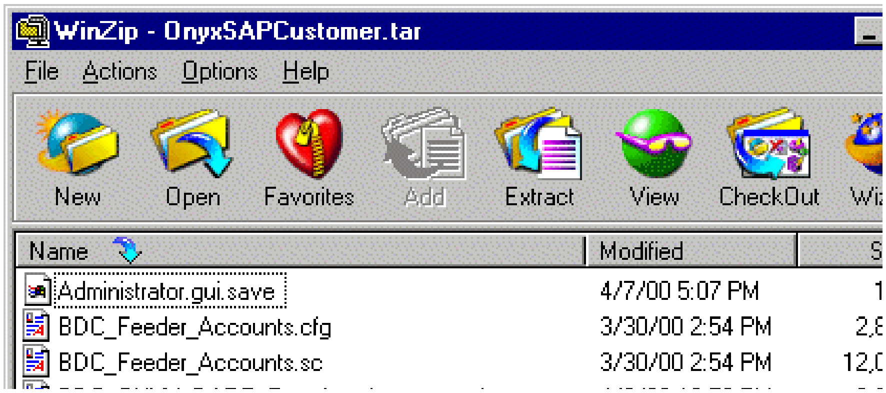
2.3.4 Schemas

After installing the Onyx/SAP Intelligent Bridge, follow the procedure below to extract, import, and commit the schema corresponding to the module you installed. If you installed more than one module, extract, import, and commit each schema separately.

1. Create a Temporary directory.
2. In the **bridges/onyx_sap** directory on the e*Gate CD, open the folder that corresponds with the module(s) you installed, and double-click the **.taz** file within that folder. For example, if you installed the SAP to Onyx Customer Bridge module,

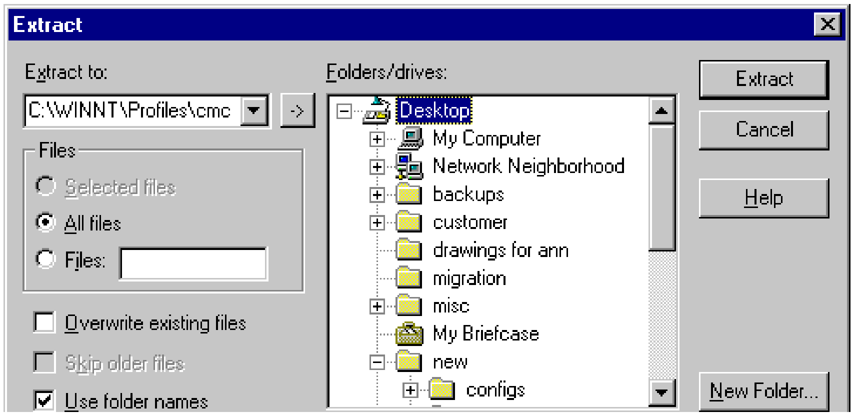
open the **bridges/onyx_sap/customers** folder and double click on the **OnyxSAPCustomer.taz** file.

3 WinZip gives you the option of decompressing the file to a temporary folder and opening it. Click **Yes**. The WinZip window opens, displaying the contents of the .taz file.



Name	Modified	S
Administrator.gui.save	4/7/00 5:07 PM	1
BDC_Feeder_Accounts.cfg	3/30/00 2:54 PM	2,8
BDC_Feeder_Accounts.sc	3/30/00 2:54 PM	12,0

4 Click the **Extract** button. The Extract window appears.



Extract to: C:\WINNT\Profiles\cmc

Files
○ Selected files
● All files
○ Files:

□ Overwrite existing files
□ Skip older files
☑ Use folder names

Folders/drives:
Desktop
My Computer
Network Neighborhood
backups
customer
drawings for ann
migration
misc
My Briefcase
new
configs

5 Select the Temporary directory you created in step 1 and click **Extract**. The following directories, as well as several other files, appear in your Temporary directory:

- configs
- monk_libraries
- monk_scripts

6 Import the schema into e*Gate by doing the following:

A Bring up a command prompt.

B Go to the Temporary directory you created in step 1.

C Run the **stcregutil.exe** utility.

D Type the following:

```
stcregutil
```

```
    -rh <hostname>
    -rs <schema_name>
    -un <user_name>
    -up <user_password>
    -i <schema_name>.schema
```

Substitute <*hostname*> with the name of the computer that is hosting the e*Gate software. Substitute <*schema_name*> with the exact name of the schema you are importing, including file extension. Enter `Administrator` and `STC` for the <*user_name*> and <*user_password*> parameters, respectively.

Note: *Be sure to import the schema that corresponds with the module you installed.*

7 Commit the **.ctl** file to e*Gate by typing the following:

```
stcregutil
-rh <hostname>
-rs <schema_name>
-un <user_name>
-up <user_password>
-ctl <schema_name>.ctl
-fc .
```

Substitute <*hostname*> with the name of the computer that is hosting the e*Gate software. Substitute <*schema_name*> with the exact name of the schema you are importing. Enter `Administrator` and `STC` for the *user_name* and *user_password* parameters, respectively. Substitute <*.ctl file*> with the exact name of the **.ctl** file that corresponds to the module you are installing.

2.3.5 e*Way Configuration

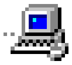
Perform the configurations listed in this section in the Components tab of the e*Gate Enterprise Manager. To open the e*Gate Enterprise Manager, do the following:

1 Click on the e*Gate Enterprise Manager icon on your desktop.

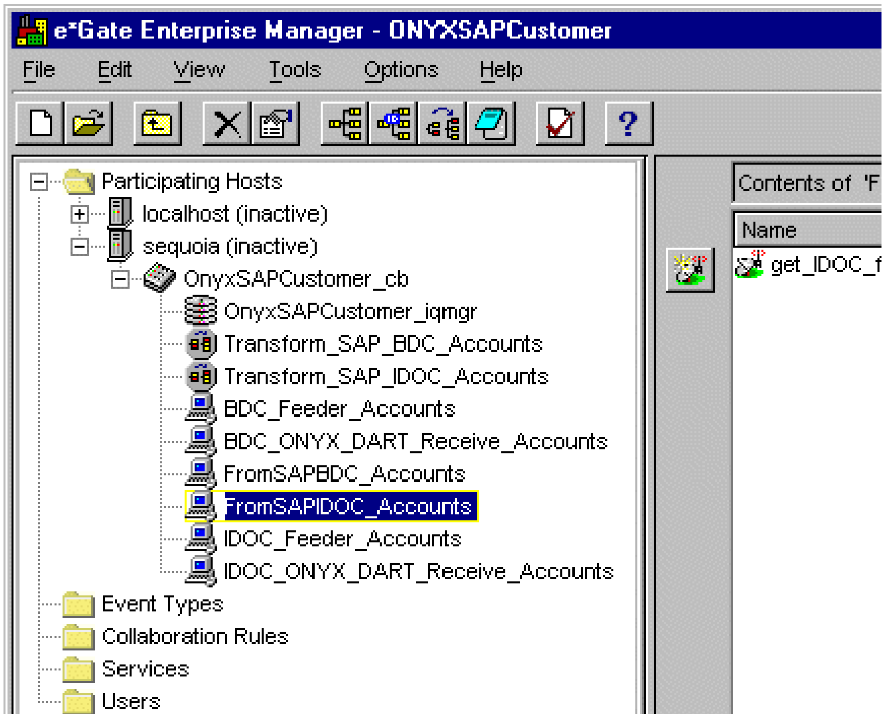
2 Enter your password (STC) and click **Open**.

3 Select the schema you wish to configure and click **Open**.

The e*Gate Enterprise Manager opens. Click on the Components tab in the lower left corner of the GUI. Then click on the plus signs (+) next to the icons until you see the e*Way icons, which look like this:



Your screen should appear as follows.



Note: *The number of instances and names of icons in the e*Gate Enterprise Manager vary depending on which schema you open.*

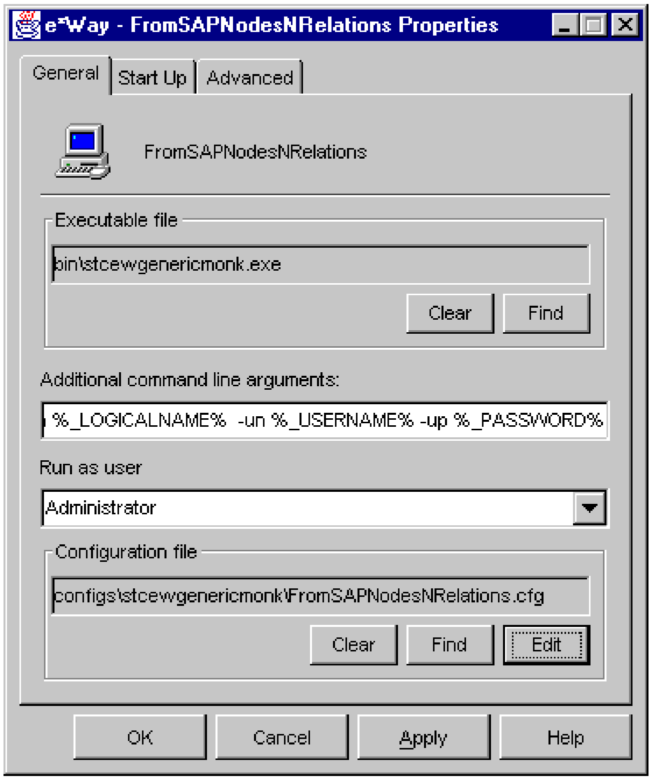
If you installed the Product module, follow the configuration instructions in **"Product Bridge" on page 16**. If you installed the Customer modules, follow the configuration instructions in **"Customer Bridge" on page 18**. If you installed both modules, follow the instructions in both sections.

See the *e*Gate System Administration and Operations Guide* for more information on using the e*Gate Enterprise Manager.

Product Bridge

Configuring the SAP BDC e*Way

1 Double-click the e*Way icon labeled **FromSAPNodesNRelations**. The **Properties** window appears.



e*Way - FromSAPNodesNRelations Properties

FromSAPNodesNRelations

Executable file: bin\stcewgenericmonk.exe

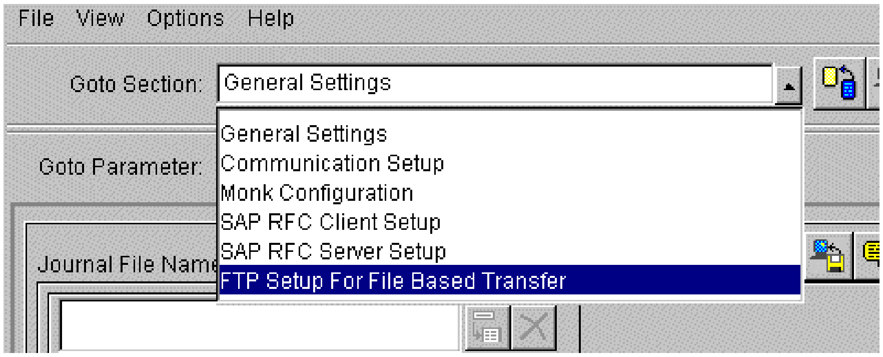
Additional command line arguments: %_LOGICALNAME% -un %_USERNAME% -up %_PASSWORD%

Run as user: Administrator

Configuration file: configs\stcewgenericmonk\FromSAPNodesNRelations.cfg

2 Click the **Edit** button. The **Edit Settings** window appears.

3 Select **FTP Setup for File Based Transfer** from the **Goto Section** drop-down list



Goto Section: General Settings

4 The parameters for the **FTP Setup for File Based Transfer** appear. Enter the appropriate data into the following fields:

- User
- Password
- Temporary File on Local System

Note: *Only enter data into the fields listed above.*

5 Select **SAP RFC Server** from the **Goto Section** drop-down list and enter the appropriate data into the following fields:

- Gateway Host Name
- Gateway Service
- Program ID

Note: *Only enter data into the fields listed above.*

6 For **SAP-BDC-READER** to run successfully, the following ABAP codes should be available in SAP system: **Z_PRD_ONYX_NODES** and **Z_PRD_ONYX_RELATIONSHIPS**.

Configuring the SAP ALE e*Way

1 Double-click the e*Way icon labeled **FromSAPMaterials**. The **Properties** window appears.

2 Click the **Edit** button. The **Edit Settings** window appears.

3 Select **ALE / RFC Setup** from the **Goto Section** and enter the appropriate data into the following fields:

- Program ID
- SAP Gateway Host
- SAP Gateway Service
- SAP System Number
- SAP System ID
- SAP Client Number
- SAP User Name

Note: *Only enter data into the fields listed above.*

4 Select **Transaction Processing** from the **Goto Section** and make the appropriate change to the Transactional ID Verification Database field.

Note: *Check **Transaction Processing** field for any other SAP ALE e*Ways installed on your system to make sure they do not use the same Transactional ID.*

Configuring the ODBC e*Ways

1 Double-click the icons with the following labels:

- **ToOnyxMaterials**
- **ToOnyxProductsGroups**
- **ToOnyxRelations**

2 The **Properties** windows appear. Click the **Edit** button on each window. The **Edit Settings** windows appear.

3 Select **Database Setup** from the **Goto Section** of each file and enter the appropriate data into the following fields:

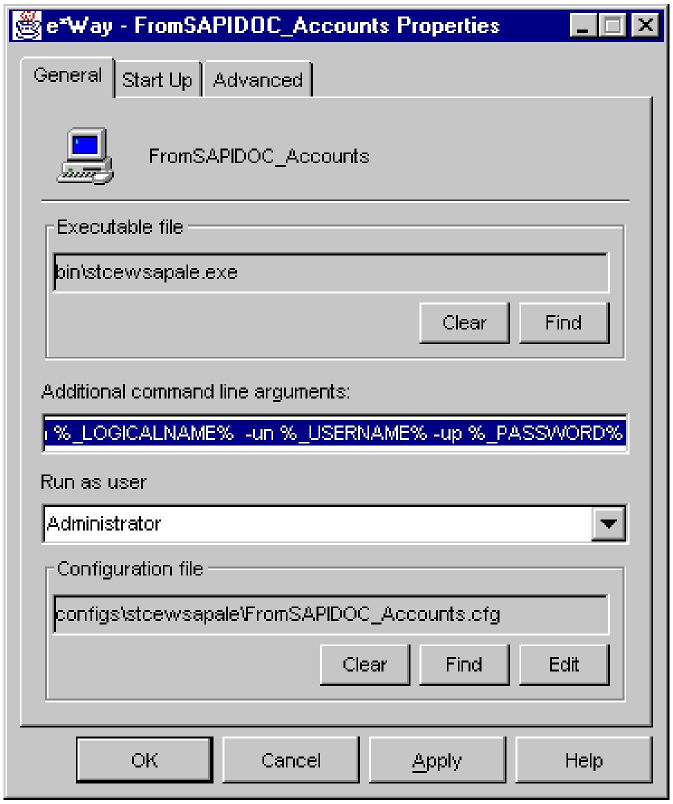
- Database Name
- Login Name
- Password

Note: *Only enter data into the fields listed above.*

Customer Bridge

Configuring the SAP ALE e*Way

1 Double-click the e*Way icon labeled **FromSAPIDOC_Accounts**. The **Properties** window appears.



Executable file: bin\stcewsapale.exe

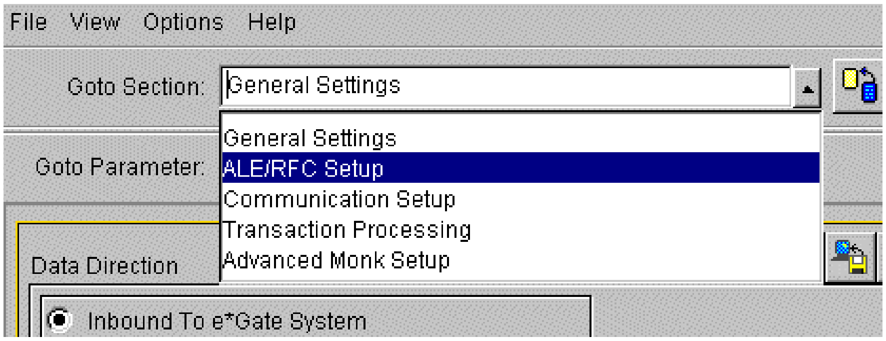
Additional command line arguments: %_LOGICALNAME% -un %_USERNAME% -up %_PASSWORD%

Run as user: Administrator

Configuration file: configs\stcewsapale\FromSAPIDOC_Accounts.cfg

2 Click the **Edit** button. The **Edit Settings** window appears.

3 Select **ALE / RFC Setup** from the **Goto Section** drop-down list.



Goto Section: General Settings

● Inbound To e*Gate System

4 Enter the appropriate data into the following fields:

- Program ID
- Gateway Host
- Gateway Service
- SAP System Number
- SAP System ID
- SAP Client Number
- SAP User Name

Note: *Only enter data into the fields listed above.*

Configuring the ODBC e*Way

1 Double-click the e*Way icon labeled **IDOC_ONYX_DART_Receive_Accounts**. The **Properties** window appears.

2 Click the **Edit** button. The **Edit Settings** window appears.

3 Select **Database Setup** from the **Goto Section** and enter the appropriate data into the following fields:

- Database Name
- Login Name
- Password

Note: *Only enter data into the fields listed above.*

4 Run the Control Broker. This starts the SAP ALE e*Way.

5 Log on to the SAP application.

6 To *add* accounts in SAP, go to transaction **VD01**. To *change* or *update* existing accounts in SAP, go to transaction **VD02**. To *delete* accounts in SAP, go to transaction **VD06**.

7 Run the change pointers **/nbd21**, type **DEBMAS**, and click the Run button on the SAP GUI to execute it.

Configuring the SAP BDC e*Way

1. Double-click the e*Way icon labeled **FromSAPBDC_Accounts**. The **Properties** window appears.
2. Click the **Edit** button. The **Edit Settings** window appears.
3. Select **SAP RFC Server** from the **Goto Section** and enter the appropriate data into the following fields:
   - Program ID
   - Gateway Host
   - Gateway Service

Note: *Only enter data into the fields listed above.*

4. Select **FTP Setup for File-Based Transfers** from the **Goto Section** and enter the appropriate data into the following fields:
   - User
   - Password
   - Temporary File on Local System
   - Temporary File on SAP Application Server

Note: *Only enter data into the fields listed above.*

5. Double-click the e*Way icon labeled **BDC_Onyx_DART_Receive_Accounts**. The **Properties** window appears.
6. Click the **Edit** button. The **Edit Settings** window appears.
7. Select **Database Setup** from the **Goto Section** and enter the appropriate data into the following fields:
   - Database Name
   - Login Name
   - Password

Note: *Only enter data into the fields listed above.*

8. Run the Control Broker (also starting the SAP BDC e*Way).
9. Log on to the SAP application.
10. Go to transaction **/NSE38** and execute transaction **Z_ACT_CH**.

2.3.6 ODBC e*Way DSN

Follow the instructions below to configure the ODBC e*Way DSN.

1. Create an ODBC Data Source Name (DSN) on the user's NT system to Onyx MS SQL Server database.

2 Change the ODBC e*Way configuration to point to the DSN, update the corresponding server name, user login, and password to DART configuration file.

Chapter 3

Module 1: The Product Bridge

3.1 Overview

The SAP to Onyx Product Bridge module synchronizes Product related data in the Onyx database to Material related data in the SAP database. This ensures that both SAP and Onyx users within a company have access to Product/Material data that is consistent, correct, and current. After installing and configuring the components of the SAP to Onyx Product Bridge module, employees using SAP can view and change (create, edit, and delete) shared Product/Material data; employees using Onyx can access shared Product/Material data in read-only format.

CAUTION: After installing and configuring the SAP to Onyx Product Bridge module, use the Onyx application to view shared data in the Product and Product Hierarchy tables. **Do not**, however, use the Onyx application to make changes to shared data in these tables. Instead, use the SAP application to make changes to shared Product/ Material data in the Material, Classification Node, and Classification Node Relationship tables in the SAP database. Changes to these SAP tables are automatically transferred to the Onyx database, enabling Onyx users to view the data. Using Onyx to manually create, edit, or delete shared Product/Material data can cause the Onyx and SAP databases to become asynchronized, defeating the purpose of the Intelligent Bridge.

To avoid accidently changing shared data in Onyx, it is recommended that you configure the Onyx database to prevent updates. For information on preventing updates in Onyx, see the documentation provided with you Onyx software.

Note: *Creates, edits, and deletes to data in the SAP database are automatically transferred to the Onyx database; however, these changes do not appear in the Onyx GUI until the Onyx application is closed and re-opened.*

3.2 Functional Description

This section describes the components and functionality of the SAP to Onyx Product Bridge.

3.2.1 Components

Material related data exist in the following SAP database tables:

- Material Master table
- Classification Relationship Node table
- Classification Node table

Product related data exist in the following Onyx database tables:

- Product Master table
- Product Hierarchy table

Communication links are established between these two sets of tables when the components of the SAP to Onyx Product Bridge module are used to create data paths. Data paths are comprised of the following components:

- SAP ALE e*Way — Extracts data from the Material Master table in IDoc format and transfers the data to an IQ.
- SAP BDC e*Way — Extracts data from the Classification Node and Classification Node Relationship tables in IDoc format, and transfers the data to two separate IQs.
- IQ (Intelligent Queue) — Stores data and routes it to the appropriate ODBC e*Way. The SAP to Onyx Product Bridge uses three separate IQs.
- ODBC e*Way — Processes data received from an IQ and transfers the data to the Onyx database by calling Onyx Product Stored Procedures. For a complete list of Onyx Product Stored Procedures, see **“Onyx Product Stored Procedures” on page 34**.

  The SAP to Onyx Product Bridge module uses three separate outbound ODBC e*Ways. Each ODBC e*Way transfers data to one of the following tables in the Onyx database:

  - Product Master table
  - Product Hierarchy table

Figure 4 on page 24 illustrates the architecture of the SAP to Onyx Product Bridge.

Figure 4 The SAP to Onyx Product Bridge Architecture

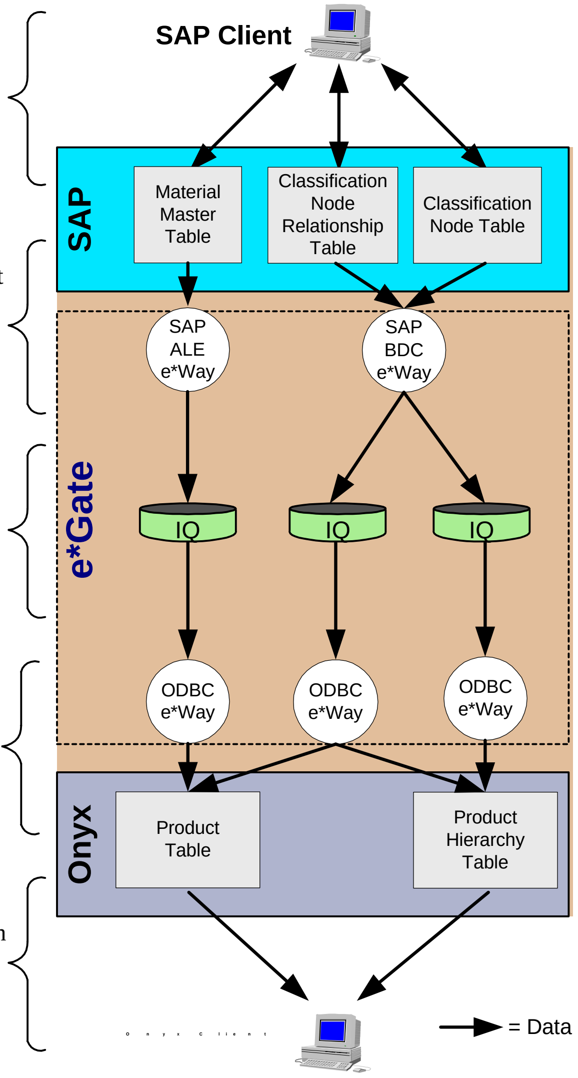
1. The SAP client is used to view and change (create, edit, and delete) data in the tables of the SAP database.

2. Inbound e*Ways extract the changed data from the SAP database in IDoc format.

3. Inbound e*Ways transfer the data to IQs, which route the data to the appropriate outbound e*Ways.

4. Outbound e*Ways translate and send the data to the appropriate Onyx database tables by calling Onyx Stored Procedures.

5. The Onyx client is used to view the data in the Onyx database. The data in these tables are now read only and synchronized with the SAP database.



3.2.2 Data Paths

After you install the SAP to Onyx Product Bridge, use the e*Gate Enterprise Manager to either implement the included SAP to Onyx Product Bridge schema, or create a schema by configuring the components installed with this module. The schema must include the following data paths:

- Material Master Table to Product Master Table
- Classification Node Table to Product Hierarchy Table
- Classification Node Relationship Table to Product and Product Hierarchy Tables

Note: *The SAP to Onyx Product Bridge module incorporates logic to only transfer changed data that are relevant to Onyx.*

Material Master Table to Product Master Table

Material Master to Product Master Data Transfer

Material records are stored in the Material Master table in the SAP database. Every time a Material record is changed (created, edited, or deleted), an IDoc message is generated by the Material Master table. An SAP ALE e*Way subscribes to the Material Master table, and uses an Event-Driven Δ Data Transfer to extract the generated IDoc message from the SAP database.

Note: *When a Material record is deleted using the SAP application, the Material record is not actually removed from the SAP database. Instead, the Material record is marked as "Deleted," and the Effective End Date of the corresponding Onyx Product record is set to the current timestamp.*

After receiving an IDoc from the Material Master table, the SAP ALE e*Way publishes the extracted data. An IQ that subscribes to the SAP ALE e*Way receives the extracted data, which it temporarily stores and publishes.

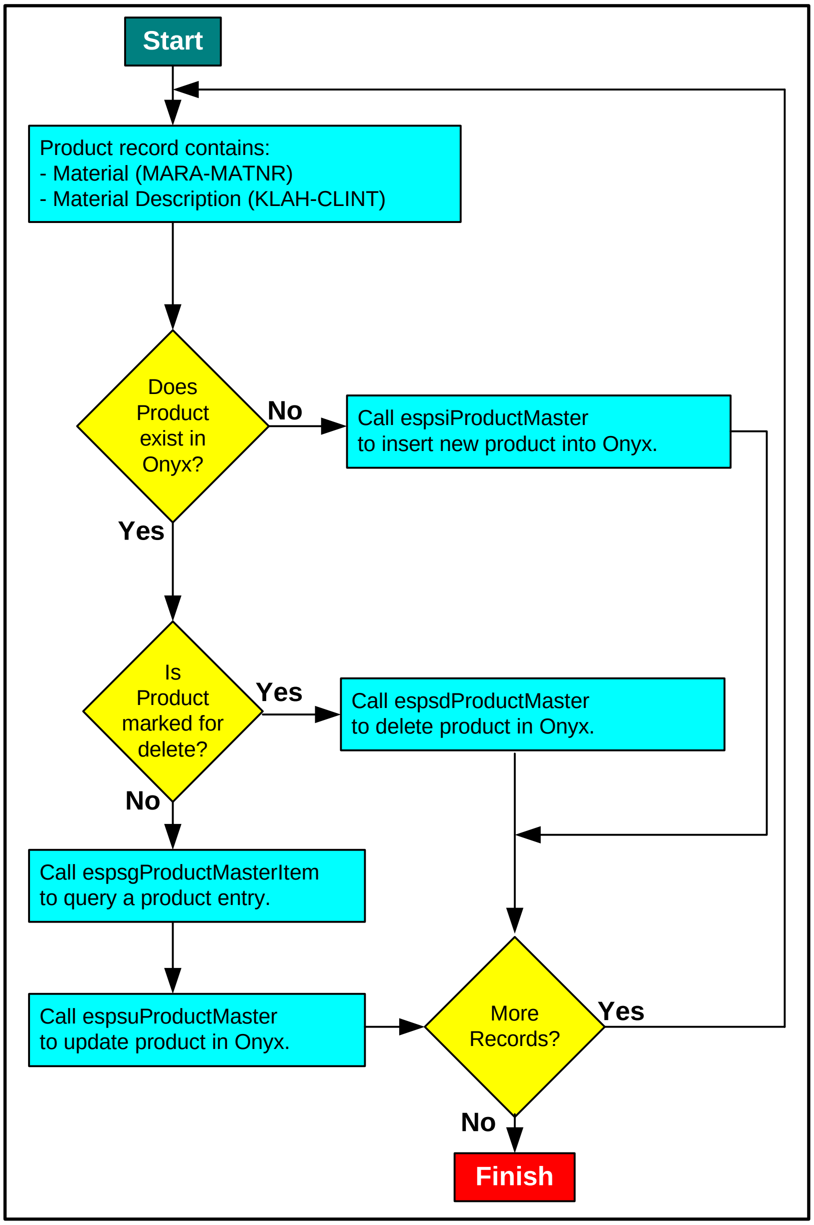
An ODBC e*Way subscribes to the IQ that receives data from the SAP ALE e*Way. The ODBC e*Way receives the data from the IQ, and places the data into the Product Master table by calling the following Onyx Product Stored Procedures:

- **espsiProductMaster** — Inserts a corresponding Product record into the Product Master table when the SAP application is used to create a new Material record.
- **espsdProductMaster** — Sets the Effective End Date of the corresponding Product record to the current timestamp when the SAP application is used to delete a Material record. This results in the expiration of the corresponding Product record in the Onyx database.
- **espsgProductMasterItem** — Finds the corresponding Product record in the Product Master table when the SAP application is used to edit a Material record.
- **espsuProductMaster** — Updates the corresponding Product record in the Product Master table when the SAP application is used to edit a Material record.

For a detailed list of Onyx Product Stored Procedures, see **"Onyx Product Stored Procedures" on page 34**.

Figure 5 on page 26 is a flowchart illustrating the logic the SAP to Onyx Product Bridge module uses to transfer Material data to the Product Master table.

Figure 5 Material to Product Master Flowchart


Start
Product record contains:
- Material (MARA-MATNR)
- Material Description (KLAH-CLINT)
Does Product exist in Onyx?
No
Call espsiProductMaster to insert new product into Onyx.
Yes
Is Product marked for delete?
Yes
Call espsdProductMaster to delete product in Onyx.
No
Call espsgProductMasterItem to query a product entry.
Call espsuProductMaster to update product in Onyx.
More Records?
Yes
No
Finish

Classification Node Table to Product Hierarchy Table

Classification Node to Product Hierarchy Data Transfer

Classification Node records are stored in the Classification Node table in the SAP database. Every time a Classification Node record is changed (created, edited, or deleted), an IDoc message is generated by the Classification Node table containing the changed data. An SAP BDC e*Way subscribes to the Material table, and uses a Batch Transfer to extract the generated IDoc message from the SAP database.

Note: *When a Classification Node record is deleted using the SAP application, the record is not actually removed from the SAP database. Instead, the record is marked as*

"Deleted," and the corresponding Product Hierarchy record in the Onyx database is set to inactive.

After receiving an IDoc from the Classification Node table, the SAP BDC e*Way publishes the extracted data. An IQ that subscribes to the SAP BDC e*Way receives the data, which it temporarily stores and publishes.

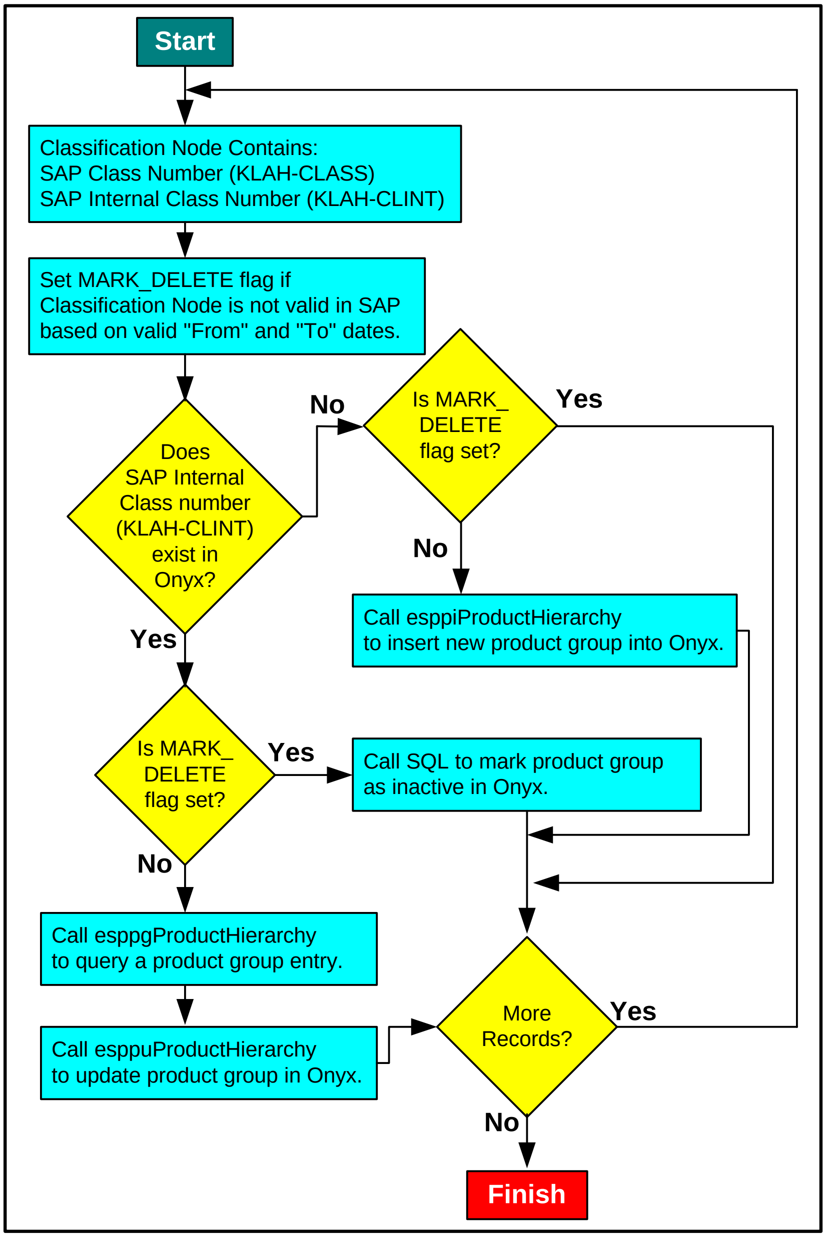
An ODBC e*Way subscribes to the IQ that receives data from the SAP BDC e*Way. The ODBC e*Way receives the data from the IQ and places the data into the Product Hierarchy table by calling the following Onyx Product Stored Procedures:

- **esppiProductHierarchy** — Inserts a corresponding Product Hierarchy record into the Product Hierarchy table when the SAP application is used to create a new Classification Node record.
- **esppgProductHierarchy** — Finds the corresponding Product Hierarchy record in the Product Hierarchy table when the SAP application is used to edit a Classification Node record.
- **esppuProductHierarchy** — Updates the corresponding Product Hierarchy record in the Onyx database when the SAP application is used to edit a Classification Node record.

For a detailed list of Onyx Product Stored Procedures, see **"Onyx Product Stored Procedures" on page 34**.

Figure 6 on page 28 is a flowchart illustrating the logic the SAP to Onyx Product Bridge module uses to transfer Classification Node data to the Product Hierarchy table.

Figure 6 Classification Node to Product Hierarchy Flowchart


Start
Classification Node Contains:
SAP Class Number (KLAH-CLASS)
SAP Internal Class Number (KLAH-CLINT)
Set MARK_DELETE flag if Classification Node is not valid in SAP based on valid "From" and "To" dates.
Does SAP Internal Class number (KLAH-CLINT) exist in Onyx?
No
Is MARK_ DELETE flag set?
Yes
No
Call esppiProductHierarchy to insert new product group into Onyx.
Yes
Is MARK_ DELETE flag set?
Yes
Call SQL to mark product group as inactive in Onyx.
No
Call esppgProductHierarchy to query a product group entry.
Call esppuProductHierarchy to update product group in Onyx.
More Records?
Yes
No
Finish

Classification Node Relationship Table to Product and Product Hierarchy Tables

Data concerning the relationships between Classification Nodes and Materials are stored in the Classification Node Relationships table in the SAP database. Classification Node Relationship records are created after both Classification Node records and Material records are created in the SAP application.

The SAP database does not store any dates associated with the Classification Node Relationship records. Therefore, the SAP application generates an IDoc that provides a "snapshot" of the Classification Node Relationship table. This IDoc is sent as a Batch data transfer, and simultaneously accounts for all changes (creates, edits, and deletes) that occur in the Classification Node Relationship table.

The generation of this IDoc is triggered manually. The IDoc must be generated when the SAP to Onyx Product Bridge is run for the first time. After the initial IDoc, trigger the generation of Classification Node Relationship IDocs whenever Classification Node Relationship records are changed (created, edited, or deleted). It is recommended that you also configure the SAP to Onyx Product Bridge to generate Classification Node Relationship IDocs on a regular schedule, such as weekly. For more information about using the SAP application to trigger IDocs, see the documentation provided with your SAP software.

Classification Node Relationship data is transferred to the Product and Product Hierarchy tables in the Onyx database. While the SAP application allows for Classification Node records and Material records to have multiple parents, the Onyx application only allows each Product record to have one parent. Therefore, in order for the SAP to Onyx Product Bridge module to function correctly, the Classification Node records and Material records in your SAP database must only have one parent.

Because the Onyx database does not allow for multiple parents in its Product Hierarchy table, the Bill of Material and Component Part data in the SAP database are not transferred to the Onyx database. Component Part data can be transferred to the Products table as Product records, but the Component Part records are not represented as components of other Products in the Onyx database. For example, a removable CD-drive may exist as a Material record in the SAP database. The CD-drive record may exist in the CD-drives Classification Node, and may be listed in the Component Parts table under Laptops. Since the CD-drive qualifies as a saleable Product, the record is transferred to the Onyx database, where it exists in the Product Hierarchy table under a Product Group called CD-drives. The CD-drive, however, would not exist in Onyx as a component of a laptop computer.

Classification Node Relationship to Product and Product Hierarchy Data Transfer

The SAP BDC e*Way that subscribes to the Classification Node table also subscribes to the Classification Node Relationship table. The SAP BDC e*Way uses a Batch Transfer to extract the IDoc containing the snapshot of the Classification Node Relationship table from the SAP database.

Note: *When a Classification Node Relationship record is deleted using the SAP application, the record is not actually removed from the SAP database. Instead, the Classification Node Relationship record is marked as "Deleted," triggering the removal of the corresponding Product record and the de-activation of the Product Hierarchy record in the Onyx database.*

After receiving an IDoc from the Classification Node Relationships table, the SAP BDC e*Way publishes the extracted data. An IQ subscribes to the SAP BDC e*Way and receives the extracted data, which it temporarily stores and publishes.

Note: *The IQ in the Classification Node Relationships to Product Hierarchy data path is not the same IQ used in the Classification Node to Product Hierarchy data path.*

Two ODBC e*Ways subscribe to the IQ that receives data from the SAP BDC e*Way.

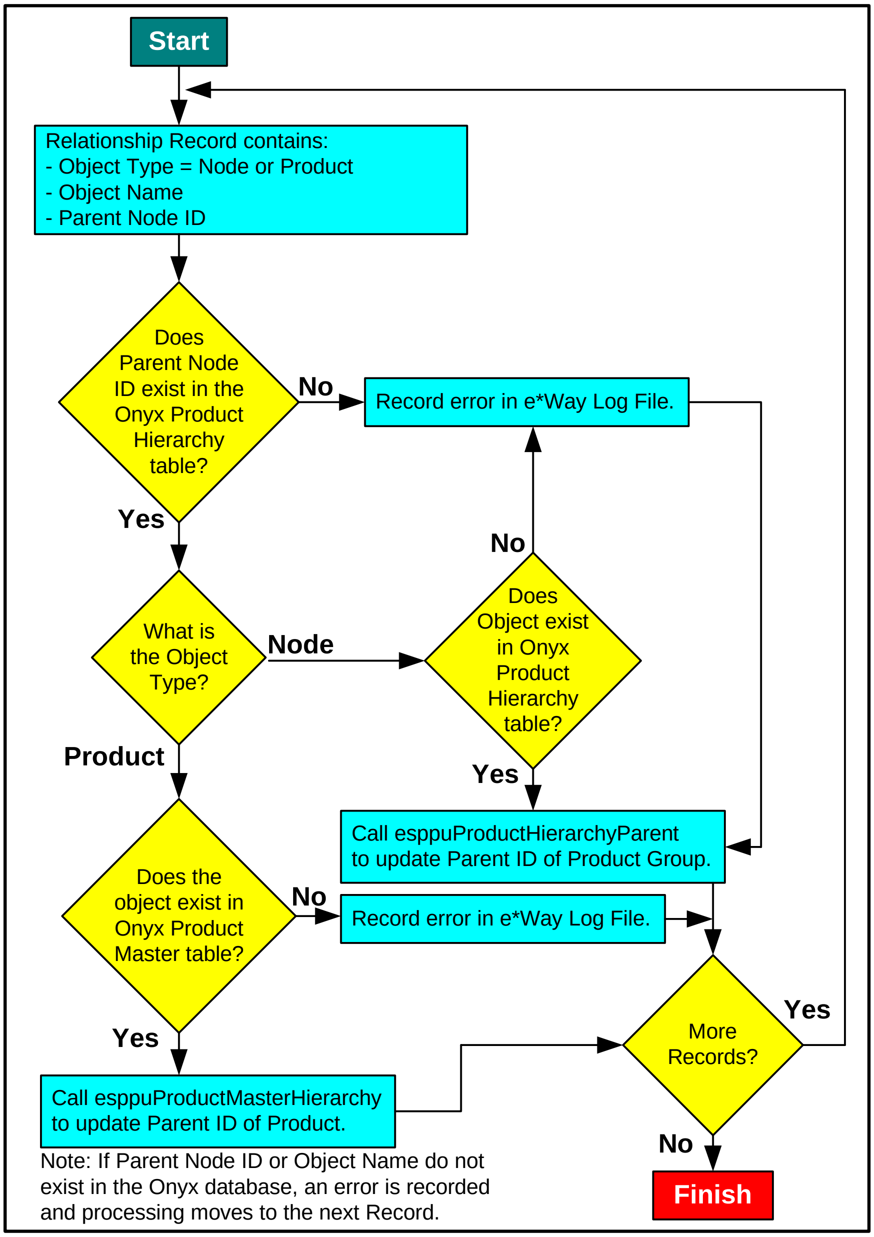
Note: *The ODBC e*Ways in the Classification Node Relationships to Product and Product Hierarchy data path are not the same ODBC e*Ways used in the Classification Node to Product Hierarchy data path.*

The ODBC e*Ways receive the extracted data from the IQ and place the data into the Product and Product Hierarchy tables by calling the following Onyx Product Stored Procedures:

- **esppuProductMasterHierarchy** — Updates the Parent ID of the corresponding Product record when the SAP application is used to edit a Classification Node Relationship record.
- **esppuProductHierarchyParent** — Updates the Parent ID of the corresponding Product Hierarchy record when the SAP application is used to edit a Classification Node Relationship record.

Figure 7 on page 31 is a flowchart illustrating the logic the SAP to Onyx Product Bridge module uses to transfer Classification Node Relationship data to the Product Hierarchy table.

Figure 7 Classification Node Relationship to Onyx Flowchart


Start
Relationship Record contains:
- Object Type = Node or Product
- Object Name
- Parent Node ID
Does Parent Node ID exist in the Onyx Product Hierarchy table?
No
Record error in e*Way Log File.
Yes
No
What is the Object Type?
Node
Does Object exist in Onyx Product Hierarchy table?
Product
Yes
Call esppuProductHierarchyParent to update Parent ID of Product Group.
Does the object exist in Onyx Product Master table?
No
Record error in e*Way Log File.
Yes
More Records?
Yes
Call esppuProductMasterHierarchy to update Parent ID of Product.
No
Finish

Note: If Parent Node ID or Object Name do not exist in the Onyx database, an error is recorded and processing moves to the next Record.

3.3 Field Mappings

The following tables provide the field and column mapping of Material/Product related data between the SAP and Onyx databases. The tables only show the data that has a suitable “match” between the two systems.

Table 2 SAP Classification Nodes/Onyx Product Hierarchy

SAP			Onyx			Comments
Field	Table - Column	Type	Field	Column	Type	
Class Number	KLAH- CLASS	CHAR 18	Description	cvhHierarchyDesc	varchar 50	Concatenate SAP class number and internal class number into Onyx description field.
Valid-to Date	KLAH- BISDT	DATS 8	Status	tiRecord Status	tinyint	Onyx only has a status indicator. If Sap Valid-to Date is today’s date or older, status is set to inactive.
Internal Class Number (SAP Key)	KLAH- CLINT	NUMC10	Description	cvhHierarchyDesc	varchar 50	Concatenate SAP class number and internal class number into Onyx description field.
Valid-from Date	KLAH- VONDT	DATS 8				Generic Onyx does not keep track of this field, however, it can be configured to do so.
"SAP Hierarchy Node"	N/A	CHAR 18				Default hard coded string

Table 3 : SAP Material Master/Onyx Product Master

SAP 4.0B			Onyx 5.0			Comments
Field	Table - Column	Type	Field	Column	Type	
Material Description	MAKT - MAKTX	CHAR 40	Description	vchDescription	varchar 50	
Material	MARA- MATNR	CHAR 18	Product Number	chProductNumber	char 20	SAP Key/Onyx Key 1
Delete Indicator	MARA- LVORM	CHAR 1				If marked with ’X,’ call esppdProductMaster.
N/A	N/A	N/A	Currency Code	chCurrencyCode	char 4	Required field in Onyx. Default to USD.
Internal Class Number of Parent Node	KLAH- CLINT	NUMC 10	Parent Internal Product Name	Parent Internal Product Name		Custom ABAP to populate this field to assign ID through UI.

3.4 Onyx Product Tables

Table 4 Onyx Product Master Table

Column Name	Type	Len	Nullable	Trim Trailing Blanks	Comments
chProductNumber	char	20	no	yes	Onyx key.
iSiteId	int	4	no	(n/a)	Onyx site identifier. Always default to ‘1’.
iAlternateLangId	int	4	no	(n/a)	
chCurrencyCode	char	4	no	yes	
vchSecondaryId	varchar	20	yes	yes	
iHierarchyId	int	4	no	(n/a)	Parent product group identifier.
vchDescription	varchar	50	no	yes	Description.
vchUser1	varchar	30	yes	yes	
vchUser2	varchar	30	yes	yes	
vchUser3	varchar	30	yes	yes	
vchUser3	varchar	30	yes	yes	
vchUser4	varchar	30	yes	yes	
vchUser5	varchar	30	yes	yes	
vchUser6	varchar	30	yes	yes	
vchUser7	varchar	30	yes	yes	
vchUser8	varchar	30	yes	yes	
vchUser9	varchar	30	yes	yes	
vchUser10	varchar	30	yes	yes	
iStockLevel	int	4	no	(n/a)	
mListPrice	money	8	yes	(n/a)	
mCost	money	8	yes	(n/a)	
vchSupportType	char	15	yes	yes	
siWarrantyCalls	smallint	2	yes	(n/a	
siWarrantyDays	smallint	2	yes	(n/a)	
dtWarrantyExpDate	datetime	8	yes	(n/a)	
vchIncidentProductNumber	varchar	20	yes	yes	
bOrderable	bit	1	no	(n/a)	
dtFirstShipDate	datetime	8	yes	(n/a)	
dtObsoleteDate	datetime	8	yes	(n/a)	
iStatusId	int	4	yes	(n/a)	
chInsertBy	char	10	no	yes	
dtInsertDate	datetime	8	no	(n/a)	
chUpdateBy	char	10	yes	yes	
dtUpdateDate	datetime	8	yes	(n/a)	
tiRecordStatus	tinyint	1	no	(n/a)	

Column Name	Type	Len	Nullable	Trim Trailing Blanks	Comments
dtModifiedDate	smalldate time	4	yes	(n/a)	

Table 5 Onyx Product Hierarchy Table

Column Name	Type	Len	Nullable	Trim Trailing Blanks	Comments
IhierarchyId	Int	4	no	(n/a)	Onyx key.
iSiteId	int	4	no	(n/a)	Onyx site identifier. Always default to ‘1’.
Ilevel	int	4	no	(n/a)	
VchHierarchyDesc	varchar	50	no	yes	Description and SAP key.
IparentID	Int	4	no	(n/a)	Parent product group identifier.
TiRecordStatus	Tinyint	1	no	(n/a)	
DtModifiedDate	small date time	4	yes	(n/a)	

3.5 Onyx Product Stored Procedures

This section contains the Onyx Product Stored Procedures used in the SAP to Onyx Product Bridge module.

Refer to the **“Onyx Product Master Table” on page 33** for more information on the Product tables.

Note: *Onyx Product Stored Procedures that start with “esp” are Onyx provided APIs.*

espsiProductMaster

Description

Inserts a new Product record by creating a new row in the Product Master table.

Example

```
espsiProductMaster 1, "1999_ACCORD_LX____", 0, "USD", null, 0,
"1999 Accord LX Sedan", null, null, null, null, null, null, null,
null, null, null, null, null, null, null, null, null, null, null,
null, null, null, null, "egate", null, null
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@chProductNumber	char	20	Maps to SAP Materials field (MARA-MATNR).
@iAlternateLangId	int	4	Onyx e*Way default this field to ‘0’.
@chCurrencyCode	char	4	Since SAP MATMAS idoc does not contain currency information, Onyx e*way default this field to ‘USD’.
@vchSecondaryId	varchar	20	
@iHierarchyId	int	4	Product parent identifier. Onyx e*Way default this field to ‘0’.
@vchDescription	varchar	50	Maps to SAP Material Description (MAKT-MAKTX)
@vchUser1	varchar	30	
@vchUser2	varchar	30	
@vchUser3	varchar	30	
@vchUser4	varchar	30	
@vchUser5	varchar	30	
@vchUser6	varchar	30	
@vchUser7	varchar	30	
@vchUser8	varchar	30	
@vchUser9	varchar	30	
@vchUser10	varchar	30	
@iStockLevel	int	4	
@mListPrice	money	8	
@mCost	money	8	
@vchSupportType	varchar	15	
@siWarrantyCalls	smallint	2	
@siWarrantyDays	smallint	2	
@dtWarrantyExpDate	datetime	8	
@vchIncidentProductNumber	varchar	20	

Parameter Name	Type	Length	Comment
@bOrderable	tinyint	1	
@dtFirstShipDate	datetime	8	
@dtObsoleteDate	datetime	8	
@iStatusId	int	4	
@chInsertBy	char	10	Onyx e*Way default this field to 'e*Gate.'
@dtInsertDate	datetime	8	
@tiRecordStatus	tinyint	1	

espsuProductMaster

Description

Updates an existing Product record by updating an existing row in the Product Master table.

Example

```
espsiProductMaster 1, "1999_ACCORD_LX____", 0, null, null, null,
"1999 Accord LX Sedan", null, null, null, null, null, null, null,
null, null, null, null, null, null, null, null, null, null, null,
null, null, null, null, "egate", null, null
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@chProductNumber	char	20	Maps to SAP Materials field (MARA-MATNR).
@iAlternateLangId	int	4	
@chCurrencyCode	char	4	
@vchSecondaryId	varchar	20	
@iHierarchyId	int	4	Product parent identifier. Onyx e*Way queries Onyx database to preserve existing value.
@vchDescription	varchar	50	Maps to SAP Material Description (MAKT-MAKTX)
@vchUser1	varchar	30	
@vchUser2	varchar	30	
@vchUser3	varchar	30	
@vchUser4	varchar	30	
@vchUser5	varchar	30	
@vchUser6	varchar	30	
@vchUser7	varchar	30	
@vchUser8	varchar	30	
@vchUser9	varchar	30	
@vchUser10	varchar	30	
@iStockLevel	int	4	
@mListPrice	money	8	
@mCost	money	8	
@vchSupportType	varchar	15	
@siWarrantyCalls	smallint	2	
@siWarrantyDays	smallint	2	
@dtWarrantyExpDate	datetime	8	
@vchIncidentProductNumber	varchar	20	
@bOrderable	tinyint	1	

Parameter Name	Type	Length	Comment
@dtFirstShipDate	datetime	8	
@dtObsoleteDate	datetime	8	
@iStatusId	int	4	
@chUpdateBy	char	10	
@dtUpdateDate	datetime	8	
@tiRecordStatus	tinyint	1	

espsdProductMaster

Description

Deletes an existing Product record by permanently removing an existing row from the Product Master table.

Example

```
esppdProductMaster 1,"1999_ACCORD_LX____",255
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@ProductNumber	char	20	Required Onyx identifier.
@RecordStatus	tinyint	1	Onyx e*Way defaults this field to 255.

espsgProductMasterItem

Description

Gets data from an existing Product record by querying the Product Master table and returning data on the specified row.

Example

```
espsqProductMasterItem 1,"1999_ACCORD_LX____"
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@ProductNumber	char	20	Required Onyx identifier.

esppiProductHierarchy

Description

Inserts a new Product Group record by creating a new row in the Product Hierarchy table.

Example

```
esppiProductHierarchy 1,0,0,1,"Honda :0000000388",1
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@HierarchyId	int	4	Onyx generated identifier.
@ParentId	int	4	For creates, Onyx e*Way defaults to ‘0’.
@Level	int	4	Onyx e*way always defaults this field to ‘1’.
@HierarchyDesc	varchar	50	Concatenate SAP classification node description (char 18) with SAP classification node identifier (char 10).
@RecordStatus	tinyint	1	‘0’ for non-active, ‘1’ for active. For creatse, Onyx e*Way defaults this field to ‘1’.

esppuProductHierarchy

Description

Updates an existing Product Group record by updating an existing row in the Product Hierarchy table.

Example

```
esppiProductHierarchy 1,9,8,1,"Honda :0000000388",1
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@HierarchyId	int	4	Required Onyx identifier.
@ParentId	int	4	For updates, Onyx e*way queries Onyx database and preserves existing value.
@Level	int	4	Onyx e*Way always defaults this field to ‘1’.
@HierarchyDesc	varchar	50	Concatenate SAP classification node description (char 18) with SAP classification node identifier (char 10).
@RecordStatus	tinyint	1	‘0’ for non-active, ‘1’ for active. For updates, Onyx e*way defaults this field to ‘1’.

esppgProductHierarchy

Description

Gets data from an existing Product Hierarchy record by querying the Product Hierarchy table and returning data on the specified row.

Example

```
esppgProductHierarchy 1,8
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@HierarchyId	int	4	Required Onyx identifier.

esppuProductMasterHierarchy

Description

Links an existing Product record to a parent Product Hierarchy record by updating an existing row in the Product Master table.

Example

```
esppuProductMasterHierarchy 1,"1999_ACCORD_LX____",9
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@ProductNumber	char	20	Required Onyx identifier.
@HierarchyId	int	4	Required parent product group identifier.

esppuProductHierarchyParent

Description

Links an existing Product Hierarchy record to a parent Product Hierarchy record by updating an existing row in the Product Hierarchy table.

Example

```
esppuProductHierarchyParent 1,12,1,9
```

Parameter List

Parameter Name	Type	Length	Comment
@SiteId	int	4	Onyx requires this to be ‘1’.
@HierarchyId	int	4	Required parent product group identifier.
@Level	int	4	Onyx e*Way defaults this field to 1.
@ParentId	int	4	Required parent product group identifier.

Chapter 4

Modules 2 and 3: The Customer Bridges

4.1 Overview

The Onyx/SAP Intelligent Bridge uses two modules to transfer Customer related data between the SAP and Onyx databases: the SAP to Onyx Customer Bridge and the Onyx to SAP Customer Bridge.

The SAP to Onyx Customer Bridge module synchronizes Customer related data in the Onyx database to Customer related data in the SAP database. The Onyx to SAP Customer Bridge module transfers Customer records created using the Onyx application to the SAP database. These two modules ensure that both SAP and Onyx users within a company have access to Customer data that is consistent, correct, and current.

After installing and configuring the Customer Bridge modules, employees using the SAP application can view, edit, and delete all shared Customer data, except for new Customer records. The SAP application can be used to edit and delete Customer records, but the Onyx application must be used to create Customer records. In addition to creating Customer records, employees using the Onyx application can access shared Customer data in read-only format. This data can be associated with Order information in the Onyx database.

CAUTION: After installing and configuring the Customer Bridge modules, use the Onyx application to create new Customer records and view shared data in the Company, Individual, and Company Hierarchy tables. **Do not**, however use the Onyx application to make changes (such as edits and deletes) to shared data in these tables. Instead, use the SAP application to make changes to the shared data in the Customer and Customer Hierarchy tables the SAP database. Changes to these SAP tables are automatically transferred to the Onyx database, enabling Onyx users to view the data. Using Onyx to manually create, edit, or delete shared Customer data can cause the Onyx and SAP databases to become asynchronized, defeating the purpose of the Intelligent Bridge.

To avoid accidently changing shared data in Onyx, it is recommended that you configure the Onyx database to prevent updates. For information on preventing updates in Onyx, see the documentation provided with you Onyx software.

Note: *Creates, edits, and deletes to data in the SAP database are automatically transferred to the Onyx database; however, these changes do not appear in the Onyx GUI until the Onyx application is closed and re-opened.*

4.2 Functional Description

This section describes the components and functionality of the Customer Bridge modules.

4.2.1 Components

Customer related data in the SAP database exist in the following SAP tables:

- Customer Master table
- Customer Hierarchy table

Customer related data in the Onyx database exist in the following tables:

- Company table
- Individual table
- Company Hierarchy table
- SAP Customer Out table (contains new Customer records that have not been exported to SAP)

Communication links are established between these two sets of tables when the components of the Customer Bridge modules are used to create data paths. Data paths are comprised of the following components:

- SAP ALE e*Way (SAP to Onyx Customer Bridge) — Extracts data from the Customer Master table in IDoc format, and transfers the data to a n IQ.
- SAP ALE e*Way (Onyx to SAP Customer Bridge) — Places new Customer records into the SAP database.
- SAP BDC e*Way — Extracts data from the Customer Hierarchy table in IDoc format, and transfers the data to an IQ.
- BOB (Business Object Broker) — Translates data and transfers it to IQs. The SAP to Onyx Customer Bridge uses two separate BOBs.

- IQ (Intelligent Queue) — Stores data and routes it to the appropriate e*Way or BOB. The SAP to Onyx Customer Bridge uses four separate IQs, and the Onyx to SAP Customer Bridge uses one.
- ODBC e*Way (SAP to Onyx Customer Bridge) — Processes data received from an IQ, and transfers the data to the Onyx database by calling Onyx Company Stored Procedures. For a complete list of Onyx Company Stored Procedures, see **"Onyx Company Stored Procedures" on page 60**.

  The SAP to Onyx Customer Bridge module uses two separate outbound ODBC e*Ways. The two ODBC e*Ways transfer data to the following Onyx database tables:

  - Company table
  - Individual table
  - Company Hierarchy table
- ODBC e*Way (Onyx to SAP Customer Bridge) — Extracts data from the SAP Customer Out table in the Onyx database, translates the data into IDoc format, and transfers the data to an IQ.

Figure 8 on page 49 illustrates the architecture of the SAP to Onyx Customer Bridge.

Figure 9 on page 50 illustrates the architecture of the Onyx to SAP Customer Bridge.

Figure 8 The SAP to Onyx Customer Bridge Architecture

1. The SAP client is used to view and change (create, edit, and delete) data in the tables of the SAP database.

2. Inbound e*Ways extract the changed data from the SAP database in IDoc format.

3. Inbound e*Ways transfer the data to IQs, which temporarily store the data

4. BOBs translate the data and send it to another IQ.

6. The IQ transfers the data to Outbound e*Ways.

7. Outbound e*Ways send the changed data to the appropriate Onyx database tables.

8. The Onyx client is used to view the data in the Onyx database. The data in these tables are now read only and synchronized with the SAP database.

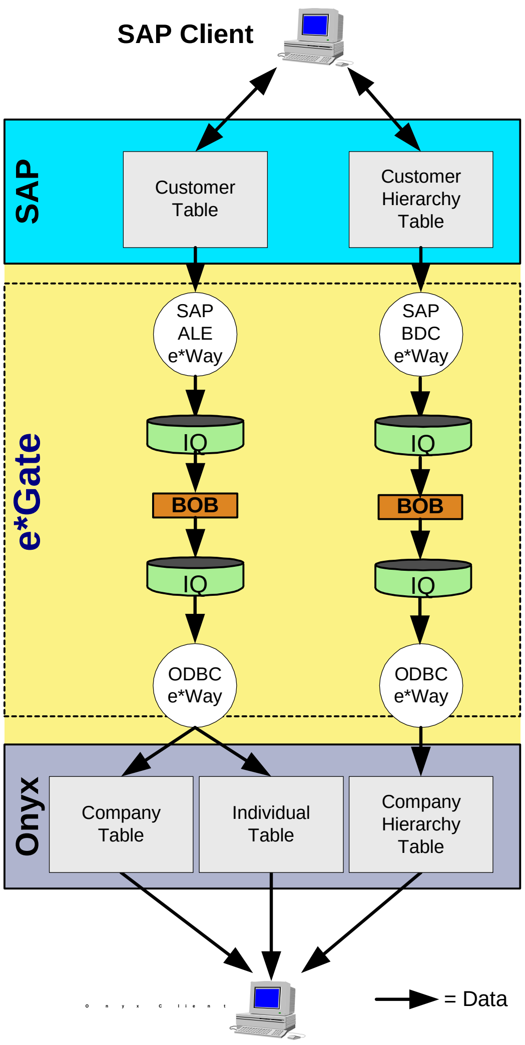
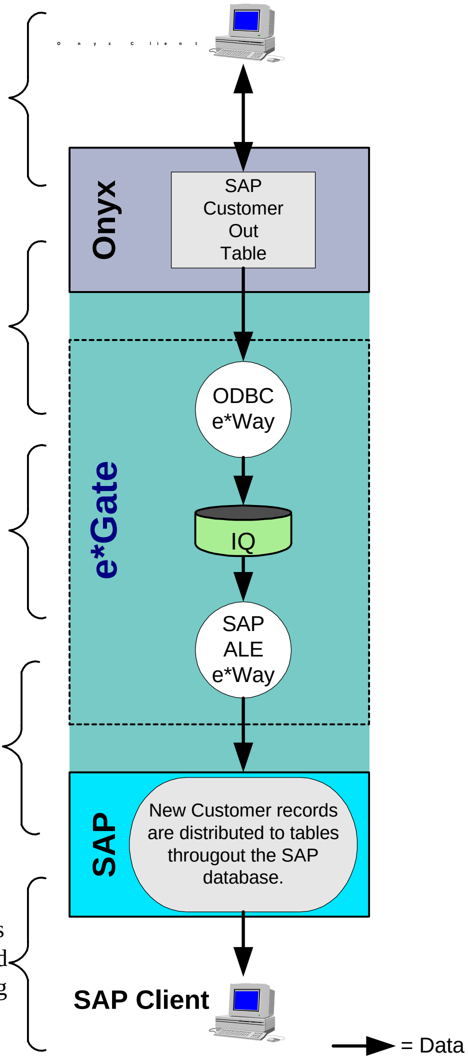


Figure 9 The Onyx to SAP Customer Bridge Architecture



1. The Onyx client is used to view and create new Customer records in the Onyx database.

2. Inbound e*Way extracts the new Customer records and changes them into IDoc format.

3. Inbound e*Way transfers the new Customer records to an IQ, which routes the data to the appropriate outbound e*Way.

4. Outbound e*Way processes and sends the changed data to the SAP database. The new Customer records are distributed to tables throughout the SAP database.

5. The new Customer records are viewed by SAP users, and can be deleted or edited using SAP, if necessary.

4.2.2 Data Paths

After you install the Customer Bridge modules, use the e*Gate Enterprise Manager to either implement the included Customer Bridge schema, or create schemas by configuring the components installed with these modules. The Customer Bridge schema includes the following data paths:

- **Customer Master Table to Company and Individual Tables** (SAP to Onyx Customer Bridge)
- **Customer Hierarchy Table to Company Hierarchy Table** (SAP to Onyx Customer Bridge)
- **SAP Customer Out Table to SAP Database** (Onyx to SAP Customer Bridge)

Note: *The Customer Bridge modules incorporate logic to only transfer changed data that are relevant to Onyx.*

Customer Master Table to Company and Individual Tables

The SAP to Onyx Customer Bridge translates SAP Customer records to Onyx Company records. While the SAP database stores various types of Customer records, the Onyx database can only store one type of Company record.

Account Types

Only the Customer records corresponding to the Account Group Types in the following table are transferred from the SAP database to the Onyx database.

Account Group Types	Customer Description
0001	Sold-To
0002	Ship-To
0003	Payer
0004	Bill-To

Each Customer record in the SAP database has one Address associated with it, so by default, that address is the Company's Primary Address in Onyx. All other types of Customer records existing in the SAP database, such as Vendors and Sales Partners, are not transferred to the Onyx database.

Partner Function Types

The SAP IDoc contains Partner Function relationship data for each Customer record. This relationship data is implemented in the Onyx database during the Event-Driven Δ Data Transfer described in "**Customer Master to Company and Individual Data Transfer**."

In the SAP database, Customer records must exist before relationships can be established. In the Onyx database, Company and Individual records must exist before the relationship is interfaced.

Only the following Partner Function Types are supported in this Intelligent Bridge.

Partner Function	Description
SH	Ship-To
PY	Payer
BP	Bill-To

Customer Master to Company and Individual Data Transfer

Customer records are stored in the Customer Master table in the SAP database. The SAP application has 4 keys that make a Customer record unique. These records are mapped to the Onyx database as follows:

- SAP Customer Name to the Onyx Company Name field in the Company Master table.

- SAP Sales Area fields (Sales Organization, Distribution Channel, and Division) to the Onyx AssignedID field in the Company Master table. (The SAP Sales Area fields are first concatenated.)

Every time a Customer record is changed (created, edited, or deleted), an IDoc message is generated by the Customer Master table. An SAP ALE e*Way subscribes to the Customer Master table, and uses an Event-Driven Δ (Delta) Data Transfer to extract the generated IDoc message from the SAP database.

Note: *When configuring the SAP ALE e*Way, be sure to configure the e*Way to extract the Partner Function Relationship data included in the IDoc generated by the Customer Master table. This data must be transferred to the Onyx database.*

Note: *When a Customer record is deleted using the SAP application, the Customer record is not actually removed from the database. Instead, the Customer record is marked as "Deleted," and the corresponding Customer record in the Onyx database is removed. Deletes of Customer Master records for all Sold-To, Bill-To, Ship-To, and Payer Customer records occur at all levels in the SAP database; the deletes of Company records occur at the Sales Area and Customer levels in the Onyx database.*

Physically removing an Customer record from the Onyx database removes all child addresses and Partner Function Relationship records, as well. This can result in other records associated with the deleted Customer record (such as Order records) referencing a record that no longer exists. Therefore, you may want to configure the Onyx application to not allow the physical removal of Customer records from the Onyx database.

Note: *Although it is possible to un-delete Customer Master records in the SAP database, the Onyx application does not support this functionality. If a record is deleted in error, a new record must be created in the SAP application and transferred to the Onyx database as a new Customer record.*

After receiving an IDoc from the Customer Master table, the SAP ALE e*Way publishes the extracted data. An IQ that subscribes to the SAP ALE e*Way receives the extracted data which it temporarily stores.

A BOB subscribes to the IQ, translates the data, and publishes it. A second IQ subscribes to the BOB. The IQ receives and publishes the data.

The ODBC e*Way subscribes to the second IQ, receives the data, and places the translated data into the Company and Individual tables by calling the following Onyx Company Stored Procedures:

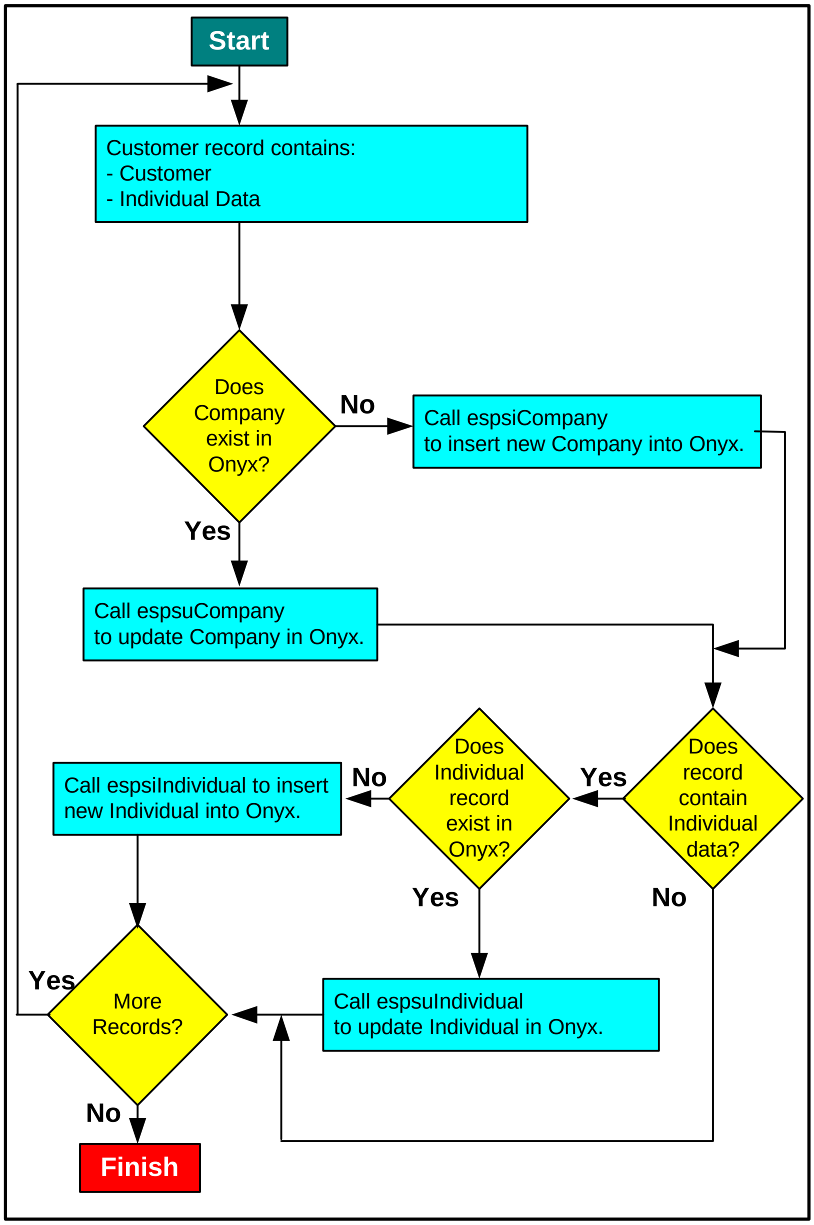
- **espsiCompany** — Inserts a corresponding Company record into the Company table when a new Customer record is created in the SAP application.
- **espsuCompany** — Updates the corresponding Company record in the Company table when a Company record is updated in the SAP application.
- **espsiIndividual** — Inserts a corresponding Individual record into the Onyx Individual table when a new Customer record is created in the SAP application.

- **espsuIndividual** — Updates the corresponding Individual record in the Individual table when a Customer record is updated in the SAP application.

For a detailed list of Onyx Company Stored Procedures, see **“Onyx Company Stored Procedures” on page 60**.

Figure 10 on page 54 is a flowchart illustrating the logic the SAP to Onyx Customer Bridge module uses to transfer Customer data to the Company and Individual tables.

Figure 10 Customer to Company and Individual Flowchart


Start
Customer record contains:
- Customer
- Individual Data
Does Company exist in Onyx?
No
Call espsiCompany to insert new Company into Onyx.
Yes
Call espsuCompany to update Company in Onyx.
Does record contain Individual data?
Yes
Does Individual record exist in Onyx?
No
Call espsiIndividual to insert new Individual into Onyx.
Yes
Call espsuIndividual to update Individual in Onyx.
No
More Records?
Yes
No
Finish

Customer Hierarchy Table to Company Hierarchy Table

Customer Hierarchy to Company Hierarchy Data Transfer

The only records in the Customer Hierarchy table that are relevant to Onyx are the Sold-To Customer Hierarchy records. Every time a Sold-To Customer Hierarchy record is changed (created, edited, or deleted), an IDoc message containing the changed data is generated by the Customer Hierarchy table. An SAP BDC e*Way subscribes to the Customer Hierarchy table and uses a Batch Transfer to extract the generated IDoc message from the SAP database.

Note: *When a Customer Hierarchy record is deleted using the SAP application, the record is not actually removed from the SAP database. Instead, the record is marked as deleted. This triggers the removal of the corresponding Company Hierarchy record in the Onyx database.*

Removing the corresponding Company Hierarchy record in the Onyx database leaves a Customer record without a Parent Customer record. Adjust your business processes if you want to avoid having "orphan" Customer records.

After receiving an IDoc from the Customer Hierarchy table, the SAP ALE e*Way publishes the extracted data. An IQ that subscribes to the SAP ALE e*Way receives the extracted data, which it publishes.

A BOB subscribes to the IQ, The BOB translates the data to ODBC format. The translated data is published to a second IQ, which temporarily stores and publishes the data.

The ODBC e*Way subscribes to the second IQ, receives the data, and places the translated data into the Company Hierarchy table by calling Onyx Company Stored Procedures.

For a detailed list of Onyx Company Stored Procedures, see **"Onyx Company Stored Procedures" on page 60**.

SAP Customer Out Table to SAP Database

New Customer Records to SAP Database Data Transfer

An ODBC e*Way subscribes to the SAP Customer Out table. The ODBC e*Way uses an Event-Driven Δ (Delta) Data Transfer to extract new Customer records from the SAP Customer Out table. After receiving the new Customer record, the ODBC translates the new Customer record to IDoc format before publishing the record.

An IQ subscribes to the ODBC e*Way and receives the translated record, which it temporarily stores and publishes.

An SAP ALE e*Way subscribes to the IQ that receives data from the ODBC e*Way. The SAP ALE e*Way takes the record and places it into the SAP database according to the instructions provided by the IDoc.

4.3 Notes

Read the following notes if you choose to create or modify schemas associated with the Customer Bridges.

- Read the **"Functional Description" on page 47** prior to creating or modifying the Customer Bridge schemas. Refer to **"Onyx Company Tables" on page 58** and **"Onyx Company Stored Procedures" on page 60** as you configure the e*Ways.

- For event-driven imports of Customer data into Onyx, use the e*Gate Universal Index to handle Account Synonyms. Even though the SAP Customer Number is stored in the Onyx Name field, the synonyms for that Customer Number can be the text name and other derivations for that Customer. This allows systems other than SAP to transfer information to Onyx without knowing the SAP Customer Number. For information on the e*Gate Universal Index, see the *e*Gate System Administration and Operations Guide.*
- Update the data-map files in e*Gate if you make any additions or updates to the language type, gender type, or address type in the Onyx application.
- When concatenating the SAP Customer Name, Sales Organization, Distribution Channel, and Division to form a unique ID, substitute spaces with a hyphen (-). SQL Server trims all spaces from varchar fields, so this is needed to preserve the uniqueness of the ID.
- If it is possible, limit the Account Types that are generated to those listed in the table **on page 51**.
- There are data being transferred from the SAP database into Account and Business Address fields in Onyx that are not defined on the vanilla user interface. The information is important and it may be necessary to expose those fields, including District in S_ADDR_ORG.COUNTY, Tax Exempt in S_ORG_EXT.TAX_EXEMP_FLG, Tax Code in S_ORG_EXT.TAX_EXEMPT_NUM.
- Because of the technical limitation in the SAP application resulting in the need to create and delete Customers and Addresses in the SAP database, it is important to remove the ability to add and delete records in any of the Customer and Business Address applets. Only allow administrators to access the corresponding Administration view. Also, implement a process for creating and deleting records.

4.4 Field Mappings

The following tables provide the field and column mapping of Customer/Account related data between the SAP and Onyx databases. The tables only show the data that has a suitable “match” between the two systems.

Table 6 SAP Customer/Onyx Company

SAP	Onyx	Comments
Field	**Field**	
Company Code	iCompanyID	
Sales Organization	iCompanyID	
Distribution Channel	iCompanyID	
Division	iSourceID	
Account Group	iCompanyID	
Name	vchCompanyName	
Search Term	iCompanyID	

SAP	Onyx	Comments
Field	Field	
City	vchCity	
Country	chCountryCode	
Language	chLanguageCode	
Postal Code	vchPostCode	
Transport Zone	iCompanyID	
Reconciliation account in GL	iCompanyID	
Shipping Conditions	iCompanyID	
Sales Tax Country	iCompanyID	
Sales Tax City	iCompanyID	
Jurisdiction Tax Code	vchTaxID	
Street	vchAddress1	
	vchAddress2	
Region	chRegionCode	
Telephone-1	vchPhoneNumber	
ContactName	vchContactFirstName	
	vchContactLastName	

Table 7 ONYX SAPCustomerOut table to SAP DEBMAS IDOC

SAP	Onyx	Comments
Field	Field	
KUNNR	iCompanyID	
SPRAS_ISO	chLanguageCode	
NAME1	vchCompanyName	
STRAS	vchAddr1Addr2	
ORT01	vchCity	
REGIO	chRegionCode	
LAND1	chCountryCode	
PSTLZ	vchPostCode	
TELF1	vchPhoneNumber	
STCD1	vchTaxID	
NAME1	vchContactName	
TELF1	vchContactPhone	
VKORG	vchSalesOrg	
VTWEG	vchDistChannel	
KTOKD	vchAcctGroup	
SORTL	vchSearchTerm	
LZONE	vchTransZone	
CITYC	vchSalesTaxCity	

4.5 Onyx Company Tables

Table 8 Onyx Company Table

Column Name	Type	Len	Nullable	Trim Trailing Blanks	Comments
iCompanyId	int	4	no	(n/a)	
iSiteIdint	int	4	no	(n/a)	
chLanguageCode	char	4	no	yes	
vchAssignedId	varchar	255	no	yes	
vchCompanyName	varchar	255	no	yes	
vchAddress1	varchar	255	yes	yes	
vchAddress2	varchar	255	yes	yes	
vchAddress3	varchar	255	yes	yes	
vchCity	varchar	255	no	yes	
chRegionCode	char	4	no	yes	
chCountryCode	char	4	no	yes	
vchPostCode	varchar	40	no	yes	
vchPhoneNumber	varchar	40	no	yes	
vchEmailAddress	varchar	255	no	yes	
vchURL	varchar	255	no	yes	
iCompanyTypeCode	int	4	yes	(n/a)	
iCompanySubTypeCode	int	4	yes	(n/a)	
iFamilyId	int	4	yes	(n/a)	
iParentId	int	4	yes	(n/a)	
iPrimaryContactId	int	4	no	(n/a)	
vchContactFirstName	varchar	255	no	yes	
vchContactLastName	varchar	255	no	yes	
iDivisionCode	int	4	yes	(n/a)	
iSICCode	int	4	yes	(n/a)	
iMarketSector	int	4	yes	(n/a)	
vchTaxId	varchar	255	yes	yes	
vchDunnsNumber	varchar	255	yes	yes	
iPhoneTypeId	int	4	yes	(n/a)	
iAddressTypeId	int	4	yes	(n/a)	
iSourceId	int	4	yes	(n/a)	
iStatusId	int	4	yes	(n/a)	
bValidAddress	tinyint	1	no	(n/a)	
iAccessCode	int	4	no	(n/a)	
bPrivate	tinyint	1	no	(n/a)	
vchUser1	varchar	255	yes	yes	
vchUser2	varchar	255	yes	yes	
vchUser3	varchar	255	yes	yes	

Column Name	Type	Len	Nullable	Trim Trailing Blanks	Comments
vchUser4	varchar	255	yes	yes	
vchUser5	varchar	255	yes	yes	
vchUser6	varchar	255	yes	yes	
vchUser7	varchar	255	yes	yes	
vchUser8	varchar	255	yes	yes	
vchUser9	varchar	255	yes	yes	
vchUser10	varchar	255	yes	yes	
chInsertBy	char	10	no	yes	
dtInsertDate	datetime	8	no	(n/a)	
chUpdateBy	char	10	no	yes	
dtUpdateDate	datetime	8	no	(n/a)	
tiRecordStatus	tinyint	1	no	(n/a)	
dtModifiedDate	smalldatetime	4	yes	(n/a)	

Table 9 Onyx Individual Table

Column Name	Type	Len	Nullable	Trim Trailing Blanks	Comments
iIndividualId	int	4	no	(n/a)	
iSiteIdint	int	4	no	(n/a)	
chLanguageCode	char	4	no	yes	
vchAssignedId	varchar	255	no	yes	
vchSalutation	varchar	255	yes	yes	
vchFirstName	varchar	255	no	yes	
vchMiddleName	varchar	255	yes	yes	
vchLastName	varchar	255	no	yes	
vchSuffix	varchar	255	yes	yes	
vchAddress1	varchar	255	yes	yes	
vchAddress2	varchar	255	yes	yes	
vchAddress3	varchar	255	yes	yes	
vchCity	varchar	255	no	yes	
chRegionCode	char	4	no	yes	
chCountryCode	char	4	no	yes	
vchPostCode	varchar	40	no	yes	
vchPhoneNumber	varchar	40	no	yes	
vchEmailAddress	varchar	255	no	yes	
vchURL	varchar	255	no	yes	
chGender	char	1	yes	yes	
iUserTypeCode	int	4	yes	(n/a)	
iUserSubTypeCode	int	4	yes	(n/a)	
iCompanyId	int	4	no	(n/a)	

Column Name	Type	Len	Nullable	Trim Trailing Blanks	Comments
vchCompanyName	varchar	255	no	yes	
chTitleCode	char	10	no	yes	
vchTitleDesc	varchar	255	no	yes	
chDepartmentCode	char	10	yes	yes	
vchDepartmentDesc	varchar	255	yes	(n/a)	
iPhoneTypeId	int	4	yes	(n/a)	
iAddressTypeId	int	4	yes	(n/a)	
iSourceId	int	4	yes	(n/a)	
iStatusId	int	4	yes	(n/a)	
bValidAddress	tinyint	1	no	(n/a)	
iAccessCode	int	4	no	(n/a)	
bPrivate	tinyint	1	no	(n/a)	
vchUser1	varchar	255	yes	yes	
vchUser2	varchar	255	yes	yes	
vchUser3	varchar	255	yes	yes	
vchUser4	varchar	255	yes	yes	
vchUser5	varchar	255	yes	yes	
vchUser6	varchar	255	yes	yes	
vchUser7	varchar	255	yes	yes	
vchUser8	varchar	255	yes	yes	
vchUser9	varchar	255	yes	yes	
vchUser10	varchar	255	yes	yes	
chInsertBy	char	10	no	yes	
dtInsertDate	datetime	8	no	(n/a)	
chUpdateBy	char	10	no	yes	
dtUpdateDate	datetime	8	no	(n/a)	
tiRecordStatus	tinyint	1	no	(n/a)	
dtModifiedDate	smalldatetime	4	yes	(n/a)	

4.6 Onyx Company Stored Procedures

This section describes the Onyx Company Stored Procedures used in the SAP to Onyx and Onyx to SAP Customer Bridge modules.

Refer to the Onyx Customer Tables for more information.

Note: *Onyx Stored Procedures that start with "esp" are Onyx provided APIs.*

espsiCompany

Description

Inserts a new Company record by creating a new row in the Company Master table.

Example

```
espsiCompany 1, NULL, ‘ENG’, ‘000000008847861212’, ‘Sesame
Grill’, ‘315 E. Huntington Dr.’, ‘Arcadia’, ‘CA’, ‘US’, ‘91006’,
‘6264716000’, NULL, NULL, NULL, NULL, NULL, NULL, NULL, '1',
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, ‘100136’, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, ‘egate’, NULL, 1
```

Parameter List

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@SiteId	int	4	Onyx requires this to be ‘1’.	
@iCompanyId	int	4	Onyx generated identifier. This is always unique.	
@chLanguageCode	char	4	e*Way uses lookup file to convert from SAP language code to ONYX language code.	KNA1-SPRAS_ISO
@vchAssignedId	varchar	255	18-digit Unique ID derived by concatenating Customer Number, Sales Organization, Distribution Channel, Division from SAP.	KNA1-KUNNR + KNVV-VKORG + KNVV-VTWEG + KNVV-SPART
@vchCompanyName	varchar	255	Mapped from SAP Customer Name	KNA1-NAME1
@vchAddress1	varchar	255	Mapped from SAP Address1	KNA1-STRAS + KNA1-PFACH
@vchAddress2	varchar	255		
@vchAddress3	varchar	255		
@vchCity	varchar	255	Mapped from SAP City	KNA1-ORT01
@chRegionCode	char	4	Mapped from SAP State	KNA1-REGIO
@chCountryCode	char	4	Mapped from SAP Country Code	KNA1-LAND1
@vchPostCode	varchar	40	Mapped from SAP Postal Code	KNA1-PSTLZ

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@vchPhoneNumber	varchar	40	Mapped from SAP Phone Number. This is the main phone number from SAP. All punctuation is filtered out.	KNA1-TELF1
@vchEmailAddress	varchar	255		
@vchURL	varchar	255		
@iCompanyTypeCode	int	4		
@iCompanySubTypeCode	int	4		
@iFamilyId	int	4		
@iParentId	int	4		
@iPrimaryContactId	int	4		
@vchContactFirstName	varchar	255		
@vchContactLastName	varchar	255		
@iDivisionCode	int	4		
@iSICode	int	4		
@iMarketSector	int	4		
@vchTaxId	varchar	255		
@vchDunnsNumber	varchar	255		
@iPhoneTypeId	int	4	Currently hard-coded to ‘100136’. Can be changed in the future to accommodate more flexibility.	
@iAddressTypeId	int	4		
@iSourceId	int	4		
@iStatusId	int	4		
@bValidAddress	tinyint	1		
@iAccessCode	int	4		
@bPrivate	tinyint	1		
@vchUser1	varchar	255		
@vchUser2	varchar	255		
@vchUser3	varchar	255		
@vchUser4	varchar	255		
@vchUser5	varchar	255		
@vchUser6	varchar	255		

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@vchUser7	varchar	255		
@vchUser8	varchar	255		
@vchUser9	varchar	255		
@vchUser10	varchar	255		
@chInsertBy	char	10	Currently hard-coded to e*Gate. Can be changed in the future to accommodate more flexibility.	
@dtInsertDate	datetime			
@tiLockRecord	tinyint	1		
@tiRecordStatus	tinyint	1		
@tireturnType	tinyint	1		

espsuCompany

Description

Updates an existing Company record by updating an existing row in the Company Master table.

Example

```
espsuCompany 1, NULL, ‘ENG’, ‘000000007745901324’, ‘Basil Thai’,
‘500 E. Huntington Dr.’, ‘Monrovia’, ‘CA’, ‘US’, ‘91006’,
‘6264716000’, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, ‘100136’, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, ‘egate’, NULL, 1
```

Parameter List

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@SiteId	int	4	Onyx requires this to be ‘1’.	
@iCompanyId	int	4		
@chLanguage Code	char	4	e*Way uses lookup file to convert from SAP language code to ONYX language code.	KNA1-SPRAS_ISO
@vchAssignedId	varchar	255	Unique ID derived by concatenating Customer Number, Sales Organization, Distribution Channel, Division from SAP.	KNA1-KUNNR + KNVV-VKORG + KNVV-VTWEG + KNVV-SPART
@vchCompany Name	varchar	255	Mapped from SAP Customer Name	KNA1-NAME1
@vchAddress1	varchar	255	Mapped from SAP Address1	KNA1-STRAS + KNA1-PFACH
@vchAddress2	varchar	255		
@vchAddress3	varchar	255		
@vchCity	varchar	255	Mapped from SAP City	KNA1-ORT01
@chRegionCode	char	4	Mapped from SAP State	KNA1-REGIO
@chCountryCode	char	4	Mapped from SAP Country Code	KNA1-LAND1
@vchPostCode	varchar	40	Mapped from SAP Postal Code	KNA1-PSTLZ

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@vchPhoneNumber	varchar	40	Mapped from SAP Phone Number. This is the main phone number from SAP. All punctuation is filtered out.	KNA1-TELF1
@vchEmailAddress	varchar	255		
@vchURL	varchar	255		
@iCompanyTypeCode	int	4		
@iCompanySubTypeCode	int	4		
@iFamilyId	int	4		
@iParentId	int	4		
@iPrimaryContactId	int	4		
@vchContactFirstName	varchar	255		
@vchContactLastName	varchar	255		
@iDivisionCode	int	4		
@iSICode	int	4		
@iMarketSector	int	4		
@vchTaxId	varchar	255		
@vchDunnsNumber	varchar	255		
@iPhoneTypeId	int	4	Currently hard-coded to ‘100136’. Can be changed in the future to accommodate more flexibility.	
@iAddressTypeId	int	4		
@iSourceId	int	4		
@iStatusId	int	4		
@bValidAddress	tinyint	1		
@iAccessCode	int	4		
@bPrivate	tinyint	1		
@vchUser1	varchar	255		
@vchUser2	varchar	255		
@vchUser3	varchar	255		

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@vchUser4	varchar	255		
@vchUser5	varchar	255		
@vchUser6	varchar	255		
@vchUser7	varchar	255		
@vchUser8	varchar	255		
@vchUser9	varchar	255		
@vchUser10	varchar	255		
@chInsertBy	char	10	Currently hard-coded to e*Gate. Can be changed in the future to accommodate more flexibility.	
@dtInsertDate	datetime			
@tiLockRecord	tinyint	1		
@tiRecordStatus	tinyint	1		
@tireturnType	tinyint	1		

espsiIndividual

Description

Deletes an existing Company record by permanently removing its row in the Company Hierarchy table.

Example

```
espsiIndividual 1, NULL, NULL, ‘0000000077’, NULL, ‘Jeffrey’,
NULL, ‘McWild’, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
‘6265551212’, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, ‘egate’, NULL, 1
```

Parameter List

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@SiteId	int	4	Onyx requires this to be ‘1’.	
@iIndividualId	int	4		
@chLanguage Code	char	4		
@vchAssignedId	varchar	255	Mapped from SAP Contact Person ID.	KNVK-PARNR
@vchSalutation	varchar	255		
@vchFirstName	varchar	255	Mapped from SAP First Name	KNVK-NAMEV
@vchMiddleName	varchar	255		
@vchLastName	varchar	255	Mapped from SAP Name	KNVK-NAME1
@vchSuffix		255		
@vchAddress1	varchar	255		
@vchAddress2	varchar	255		
@vchAddress3	varchar	255		
@vchCity	varchar	255		
@chRegionCode	char	4		
@chCountryCode	char	4		
@vchPostCode	varchar	40		
@vchPhoneNumber	varchar	40	Mapped from SAP Telephone. All punctuation is filtered out.	KNVK-TELF1
@vchEmailAddress	varchar	255		
@vchURL	varchar	255		

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@chGender	char	1		
@iUserTypeId	int	4		
@iUserSubTypeId	int	4		
@iCompanyId	int	4		
@vchCompanyName	varchar	255		
@chTitleCode	char	10		
@vchTitleDesc	varchar	255		
@chDepartmentCode	char	10		
@vchDepartmentDesc	varchar	255		
@iPhoneTypeId	int	4		
@iAddressTypeId	int	4		
@iSourceId	int	4		
@iStatusId	int	4		
@bValidAddress	tinyint	1		
@iAccessCode	int	4		
@bPrivate	tinyint	1		
@vchUser1	varchar	255		
@vchUser2	varchar	255		
@vchUser3	varchar	255		
@vchUser4	varchar	255		
@vchUser5	varchar	255		
@vchUser6	varchar	255		
@vchUser7	varchar	255		
@vchUser8	varchar	255		
@vchUser9	varchar	255		
@vchUser10	varchar	255		
@chInsertBy	char	10	Currently hard-coded to e*Gate. Can be changed in the future to accommodate more flexibility.	
@dtInsertDate	datetime			
@tiLockRecord	tinyint	1		
@tiRecordStatus	tinyint	1		
@tireturnType	tinyint	1		

espsuIndividual

Description

Updates data in an existing Individual record by updating an existing row in the Individual table.

Example

```
espsuIndividual 1, NULL, NULL, ‘0000000077’, NULL, ‘Jeff’, NULL,
‘McWildy’, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
‘6265551212’, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, ‘egate’, NULL, 1
```

Parameter List

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@SiteId	int	4	Onyx requires this to be ‘1’.	
@iIndividualId	int	4		
@chLanguage Code	char	4		
@vchAssignedId	varchar	255	Mapped from SAP Contact Person ID.	KNVK-PARNR
@vchSalutation	varchar	255		
@vchFirstName	varchar	255	Mapped from SAP First Name	KNVK-NAMEV
@vchMiddleName	varchar	255		
@vchLastName	varchar	255	Mapped from SAP Name.	KNVK-NAME1
@vchSuffix		255		
@vchAddress1	varchar	255		
@vchAddress2	varchar	255		
@vchAddress3	varchar	255		
@vchCity	varchar	255		
@chRegionCode	char	4		
@chCountryCode	char	4		
@vchPostCode	varchar	40		
@vchPhoneNumber	varchar	40	Mapped from SAP Telephone. All punctuation is filtered out.	KNVK-TELF1
@vchEmailAddress	varchar	255		

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@vchURL	varchar	255		
@chGender	char	1		
@iUserTypeId	int	4		
@iUserSubTypeId	int	4		
@iCompanyId	int	4		
@vchCompanyName	varchar	255		
@chTitleCode	char	10		
@vchTitleDesc	varchar	255		
@chDepartmentCode	char	10		
@vchDepartmentDesc	varchar	255		
@iPhoneTypeId	int	4		
@iAddressTypeId	int	4		
@iSourceId	int	4		
@iStatusId	int	4		
@bValidAddress	tinyint	1		
@iAccessCode	int	4		
@bPrivate	tinyint	1		
@vchUser1	varchar	255		
@vchUser2	varchar	255		
@vchUser3	varchar	255		
@vchUser4	varchar	255		
@vchUser5	varchar	255		
@vchUser6	varchar	255		
@vchUser7	varchar	255		
@vchUser8	varchar	255		
@vchUser9	varchar	255		
@vchUser10	varchar	255		
@chInsertBy	char	10	Currently hard-coded to e*Gate. Can be changed in the future to accommodate more flexibility.	
@dtInsertDate	datetime			
@tiLockRecord	tinyint	1		
@tiRecordStatus	tinyint	1		

Parameter Name	Type	Length	Comment	SAP IDOC Segment-Node
@tireturnType	tinyint	1		

espsuCompanyBatch

Description

Updates an existing Company Hierarchy record in the Company table.

Example

```
espsuCompanyBatch 1, '14', NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, '1',
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, 'egate', NULL, 1
```

Parameter List

Parameter Name	Type	Length	Comment	SAP IDOC Segment -Node
@SiteId	int	4	Onyx requires this to be '1'.	
@iCompanyId	int	4	DART e*Way does a lookup on the Company table based on the concatenated SAP fields and retrieves the child CompanyId.	
@chLanguageCode	char	4		
@vchAssignedId	varchar	255		
@vchCompanyName	varchar	255		
@vchAddress1	varchar	255		
@vchAddress2	varchar	255		
@vchAddress3	varchar	255		
@vchCity	varchar	255		
@chRegionCode	char	4		
@chCountryCode	char	4		
@vchPostCode	varchar	40		
@vchPhoneNumber	varchar	40		
@vchEmailAddress	varchar	255		
@vchURL	varchar	255		
@iCompanyTypeCode	int	4		

Parameter Name	Type	Length	Comment	SAP IDOC Segment -Node
@iCompanySubTypeCode	int	4		
@iFamilyId	int	4		
@iParentId	int	4		
@iPrimaryContactId	int	4		
@vchContactFirstName	varchar	255		
@vchContactLastName	varchar	255		
@iDivisionCode	int	4		
@iSICode	int	4		
@iMarketSector	int	4		
@vchTaxId	varchar	255		
@vchDunnsNumber	varchar	255		
@iPhoneTypeId	int	4	Currently hard-coded to ‘100136’. Can be changed in the future to accommodate more flexibility.	
@iAddressTypeId	int	4		
@iSourceId	int	4		
@iStatusId	int	4		
@bValidAddress	tinyint	1		
@iAccessCode	int	4		
@bPrivate	tinyint	1		
@vchUser1	varchar	255		
@vchUser2	varchar	255		
@vchUser3	varchar	255		
@vchUser4	varchar	255		
@vchUser5	varchar	255		
@vchUser6	varchar	255		
@vchUser7	varchar	255		
@vchUser8	varchar	255		
@vchUser9	varchar	255		
@vchUser10	varchar	255		

Parameter Name	Type	Length	Comment	SAP IDOC Segment -Node
@chInsertBy	char	10	Currently hard-coded to e*Gate. Can be changed in the future to accommodate more flexibility.	
@dtInsertDate	datetime			
@tiLockRecord	tinyint	1		
@tiRecordStatus	tinyint	1		
@tireturnType	tinyint	1		

Index