Oracle® Retail Strategic Store Solutions

Implementation Guide, Volume 2 – Oracle Retail Strategic Store Solutions Implementation Solutions

Release 13.2

April 2010

ORACLE®

Oracle Retail Strategic Store Solutions Implementation Solutions, Release 13.2

Primary Author: Graham Fredrickson

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server - Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning, Oracle Retail Demand Forecasting, Oracle Retail Regular Price Optimization, Oracle Retail Size Profile Optimization, Oracle Retail Replenishment Optimization applications.

(ii) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

(iv) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by SAP and imbedded in Oracle Retail Store Inventory Management.

(vi) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(vii) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

List of Examples

List of Figures

Send Us Your Comments

Oracle Retail Strategic Store Solutions Implementation Guide, Volume 2, Release 13.2

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Audience

The Implementation Guide is intended for the Oracle Retail Point-of-Service integrators and implementation staff, as well as the retailer's IT personnel.

Related Documents

For more information, see the following documents in the Oracle Retail Release 13.2 documentation set:

- Oracle Retail Strategic Store Solutions Licensing Information
- Oracle Retail Back Office documentation set
- Oracle Retail Labels and Tags documentation set
- Oracle Retail Central Office documentation set
- Oracle Retail Point-of-Service documentation set

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

- https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to recreate
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

If you are installing the application for the first time, you install either a base release (for example, 13.2) or a later patch release (for example, 13.2.2). If you are installing a software version other than the base release, be sure to read the documentation for each patch release (since the base release) before you begin installation. Patch documentation can contain critical information related to the base release and code changes that have been made since the base release.

Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site (with the exception of the Data Model which is only available with the release packaged code):

http://www.oracle.com/technology/documentation/oracle_retail.html

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

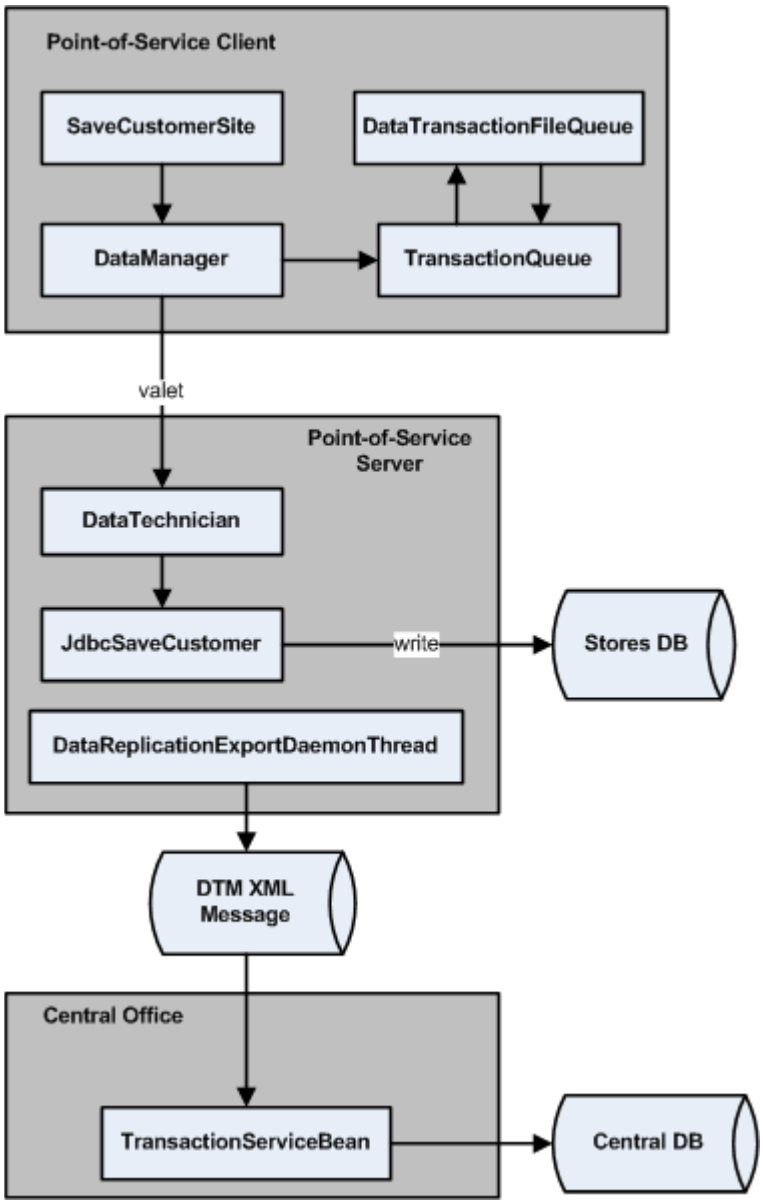
Centralized Customer

Centralized Customer allows an Oracle Retail Central Office operator to enter and manage customer data. Centralized Customer also provides Oracle Retail Point-of-Service the ability to retrieve customer information from a central database. This functionality enables Point-of-Service to support customer-specific pricing. Other Point-of-Service features support the need for Centralized Customer such as assisting in pickup and delivery orders, and obtaining Tax ID numbers for customers required to manage specific tax forms. Retail stores and cashiers also benefit from this functionality. Since customer information can be retrieved from a central database, customer information does not have to be re-entered at different stores.

The Centralized Customer package enables the operator to manage existing customer information and add new customers to the central database. The operator has the ability to search for a customer, modify existing customer information, or mark a customer's record for deletion from the database. The operator can also assign a Pricing Group to the customer which allows retailers the ability to offer customer specific pricing. Pricing Groups can be assigned to a Price Promotion or Discount Rule.

There are two types of customers: Individual and Business. Business customers require slightly different data than individual customers such as tax certificate numbers.

Figure 1–1 Centralized Customer Object Model



2

Changing and Configuring a New Base Currency

Changing Currency

In order to switch to another base and alternate currency, perform the following steps:

1. Set the base currency flag in the primary currency of the currency table. For example, if EUR is the base currency:

   ```
   update co_cny set FL_CNY_BASE='1' where DE_CNY='EUR'
   ```

2. Remove the base currency flag from any other currencies in that table. For example:

   ```
   update co_cny set FL_CNY_BASE = '0' where DE_CNY <> 'EUR'
   ```

3. Enforce ordering so that the primary currency is first and the alternate currency is second for the AI_CNY_PRI column in the currency table. Other rows should be ordered, but the specific order isn't important. For example if EUR is base currency and GBP is the alternate:

   ```
   update co_cny set AI_CNY_PRI=0 where DE_CNY='EUR'
   update co_cny set AI_CNY_PRI=1 where DE_CNY='GBP'
   update co_cny set AI_CNY_PRI=2 where DE_CNY='USD'
   update co_cny set AI_CNY_PRI=3 where DE_CNY='CAD'
   update co_cny set AI_CNY_PRI=4 where DE_CNY='MXN'
   update co_cny set AI_CNY_PRI=5 where DE_CNY='JPY'
   ```

4. Add store safe tenders supported for the new base/alternate currency. For example, if EUR is the new base currency, add money order tender support for EUR:

   ```
   insert into le_tnd_str_sf
   (ID_RPSTY_TND, TY_TND, TY_SB_TND, LU_CNY_ISSG_CY, TS_CRT_RCRD, TS_MDF_RCRD, ID_
   CNY_ICD )
   VALUES ('1','MNYO', ' ', 'EU', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, 5);
   ```

   Remove store safe tenders no longer supported for the old base/alternate currency. For example, if USD if the old base currency, remove money order tender support for USD:

   ```
   delete from le_tnd_str_sf where LU_CNY_ISSG_CY = 'US' and TY_TND = 'MNYO';
   ```

5. Add exchange rate records for alternate and base currencies into the CO_RT_EXC table based on the new base currency. Delete all exchange rate records based on any previous base currency.

There are some application parameters that must be changed as well:

- Tender Group:
  - `CashAccepted`: For example, if EUR is base and GBP is alternate, make sure that the `CashAccepted` parameter is changed so that EUR and GBP are selected.
  - `TravelersChecksAccepted`: For EUR as base and GBP as alternate, the values for the `TravelersChecksAccepted` parameter should be EURCHK and GBPCHK.
  - `ChecksAccepted`: For EUR as base and GBP as alternate, the values for the `ChecksAccepted` parameter should be EURCHK and GBPCHK.
  - `GiftCertificateAccepted`: Change the values to reflect all the currencies accepted (base and alternate). For example the values may be EUR and GBP, or EUR, GBP and USD.
  - `StoreCreditAccepted`: Change the values to reflect all the currencies accepted (base and alternate). For example the values may be EUR and GBP, or EUR, GBP and USD.
- Reconciliation Group:
  - `TendersToCountAtTillReconcile`: For EUR as base and GBP as alternate, the values for the `TendersToCountAtTillReconcile` parameter should be:
    * Cash
    * Check
    * ECheck
    * Credit
    * Debit
    * TravelCheck
    * GiftCert
    * Coupon
    * GiftCard
    * StoreCredit
    * MallCert
    * PurchaseOrder
    * MoneyOrder
    * GBPCash
    * GBPTravelCheck
    * GBPCheck
    * GBPGiftCert
    * GBPStoreCredit

Configuring a New Base Currency

Throughout this section, Krona is used as the example new base currency that is being configured. The Krona currency code is SEK, and the issuing country code is SE.

Currency SQL Configuration

The following SQL configurations for Currency are available.

Currency Table CO_CNY

A new record describing the new currency information such as its currency code, issuing country code and so forth, must be inserted into this table.

In the base currency flag column **FL_CNY_BASE**, the new currency must be set to **1** indicating that it is the base. The flag for other currencies must be set to **0**, indicating that they are alternate currencies.

> **Note:** Point-of-Service supports base-plus-one alternate currency. The priority column AI_CNY_PRI must be set to 0 for the new base currency. It must be set to 1 for the supported alternate currency. For other alternate currencies, they must be ordered and greater than 1, but the specific order isn't important.

Example 2–1 Add Krona as Base to Currency Table CO_CNY

```
INSERT INTO CO_CNY
(ID_CNY_ICD, LU_CNY_ISSG_CY, CD_CNY_ISO, DE_CNY, DE_CNY_ISSG_NAT, FL_CNY_BASE, QU_
CNY_SCLE, AI_CNY_PRI)
VALUES (7,'SE', 'SEK', 'SEK', 'Sweden', '1', 2, 0);

UPDATE CO_CNY
SET FL_CNY_BASE = '0'
WHERE CD_CNY_ISO <> 'SEK';

UPDATE CO_CNY
SET AI_CNY_PRI = AI_CNY_PRI + 1
WHERE CD_CNY_ISO <> 'SEK';
```

Currency Denomination Table CO_CNY_DNM and I8 table CO_CNY_DNM_I8

Denominations for the new base currency must be added to the CO_CNY_DNM and CO_CNY_DNM_I8 table. For example:

Example 2–2 Add Krona Denominations to Denomination Table CO_CNY_DNM

```
INSERT INTO CO_CNY_DNM
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 1, 'SE_50Ores', '0.50', 1);

INSERT INTO CO_CNY_DNM
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 2, 'SE_1Kronas', '1.00', 2);

INSERT INTO CO_CNY_DNM
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 3, 'SE_5Kronas', '5.00', 3);

INSERT INTO CO_CNY_DNM
```

```
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 4, 'SE_10Kronas', '10.00', 4);

INSERT INTO CO_CNY_DNM
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 5, 'SE_20Kronas', '20.00', 5);

INSERT INTO CO_CNY_DNM
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 6, 'SE_50Kronas', '50.00', 6);

INSERT INTO CO_CNY_DNM
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 7, 'SE_100Kronas', '100.00', 7);

INSERT INTO CO_CNY_DNM
(ID_CNY_ICD, ID_CNY_DNM, NM_DNM, VL_DNM, CD_DNM_DPLY_PRI)
VALUES (7, 8, 'SE_1000Kronas', '1000.00', 8);
```

Example 2–3 Add Krona Denominations to I8 Table CO_CNY_DNM_I8

```
INSERT INTO CO_CNY_DNM_I8
(ID_CNY_ICD, ID_CNY_DNM, LCL, NM_DNM)
VALUES (7, 2,'en','1 Kronas');

INSERT INTO CO_CNY_DNM_I8
(ID_CNY_ICD, ID_CNY_DNM, LCL, NM_DNM)
VALUES (7, 2,'fr','1 couronne');
```

Note: For each denomination record in the CON_CNY_DNM table, there are I8 records in CO_CNY_DNM_I8 table, one for each supported language.

Exchange Rate Table CO_RT_EXC

Add exchange rate records for alternate and base currencies into the CO_RT_EXC table based on the new base currency. Delete all exchange rate records based on any previous base currency. For example:

Example 2–4 Add Alternate Currency Exchange Rates to Krona

```
-- Delete all the existing records
Delete from CO_RT_EXC;

INSERT INTO CO_RT_EXC
(LL_CNY_EXC, DC_RT_EXC_EF, DC_RT_EXC_EP, ID_CNY_ICD, MO_RT_TO_BUY, MO_RT_TO_SL,
MO_FE_SV_EXC)
VALUES(0.00, TO_DATE('1990-01-01', 'YYYY-MM-DD'), TO_DATE('2099-12-31',
'YYYY-MM-DD'), 1, 6.3337, 6.3362, 0.00);

INSERT INTO CO_RT_EXC
(LL_CNY_EXC, DC_RT_EXC_EF, DC_RT_EXC_EP, ID_CNY_ICD, MO_RT_TO_BUY, MO_RT_TO_SL,
MO_FE_SV_EXC)
VALUES(0.00, TO_DATE('1990-01-01', 'YYYY-MM-DD'), TO_DATE('2099-12-31',
'YYYY-MM-DD'), 2, 6.2849, 6.2898, 0.00);

INSERT INTO CO_RT_EXC
(LL_CNY_EXC, DC_RT_EXC_EF, DC_RT_EXC_EP, ID_CNY_ICD, MO_RT_TO_BUY, MO_RT_TO_SL,
```

```
MO_FE_SV_EXC)
VALUES(0.00, TO_DATE('1990-01-01', 'YYYY-MM-DD'), TO_DATE('2099-12-31',
'YYYY-MM-DD'), 3, 0.5799, 0.5816, 0.00);

INSERT INTO CO_RT_EXC
(LL_CNY_EXC, DC_RT_EXC_EF, DC_RT_EXC_EP, ID_CNY_ICD, MO_RT_TO_BUY, MO_RT_TO_SL,
MO_FE_SV_EXC)
VALUES(0.00, TO_DATE('1990-01-01', 'YYYY-MM-DD'), TO_DATE('2099-12-31',
'YYYY-MM-DD'), 4, 12.434, 12.441, 0.00);

INSERT INTO CO_RT_EXC
(LL_CNY_EXC, DC_RT_EXC_EF, DC_RT_EXC_EP, ID_CNY_ICD, MO_RT_TO_BUY, MO_RT_TO_SL,
MO_FE_SV_EXC)
VALUES(0.00, TO_DATE('1990-01-01', 'YYYY-MM-DD'), TO_DATE('2099-12-31',
'YYYY-MM-DD'), 5, 9.3739, 9.3796, 0.00);

INSERT INTO CO_RT_EXC
(LL_CNY_EXC, DC_RT_EXC_EF, DC_RT_EXC_EP, ID_CNY_ICD, MO_RT_TO_BUY, MO_RT_TO_SL,
MO_FE_SV_EXC)
VALUES(0.00, TO_DATE('1990-01-01', 'YYYY-MM-DD'), TO_DATE('2099-12-31',
'YYYY-MM-DD'), 6, 0.05782, 0.05786, 0.00);

INSERT INTO CO_RT_EXC
(LL_CNY_EXC, DC_RT_EXC_EF, DC_RT_EXC_EP, ID_CNY_ICD, MO_RT_TO_BUY, MO_RT_TO_SL,
MO_FE_SV_EXC)
VALUES(0.00, TO_DATE('1990-01-01', 'YYYY-MM-DD'), TO_DATE('2099-12-31',
'YYYY-MM-DD'), 7, 1.0, 1.0, 0.00);
```

Store Safe Tender Table LE_TND_STR_SF

Add the store safe tenders supported for the new base currency. For example:

Example 2–5 Add Store Safe Tenders for Krona

```
INSERT INTO LE_TND_STR_SF
    ( ID_RPSTY_TND, TY_TND, TY_SB_TND, LU_CNY_ISSG_CY, TS_CRT_RCRD, TS_MDF_RCRD,
ID_CNY_ICD )
    VALUES('1','CASH', ' ', 'SE', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, 7);
INSERT INTO LE_TND_STR_SF
    ( ID_RPSTY_TND, TY_TND, TY_SB_TND, LU_CNY_ISSG_CY, TS_CRT_RCRD, TS_MDF_RCRD,
ID_CNY_ICD )
    VALUES('1','CHCK', ' ', 'SE', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, 7);
INSERT INTO LE_TND_STR_SF
    ( ID_RPSTY_TND, TY_TND, TY_SB_TND, LU_CNY_ISSG_CY, TS_CRT_RCRD, TS_MDF_RCRD,
ID_CNY_ICD )
    VALUES('1','TRAV', ' ', 'SE', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, 7);

-- MoneyOrderSafeTender

INSERT INTO LE_TND_STR_SF
(ID_RPSTY_TND, TY_TND, TY_SB_TND, LU_CNY_ISSG_CY, TS_CRT_RCRD, TS_MDF_RCRD, ID_
CNY_ICD )
VALUES ('1','MNYO', ' ', 'SE', CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, 7);
```

Money Order Tenders are only accepted for base currency, therefore before inserting records for the new base currency, delete any money order tenders for the other currencies:

```
DELETE * from LE_TND_STR_SF where ty_tnd='MNYO'
```

Parameter Configuration

The following tender parameters must be enhanced to include the new base currency:

- StoreCreditsAccepted
- ChecksAccepted
- CashAccepted
- GiftCertificatesAccepted
- TravelersChecksAccepted

The reconciliation parameter **TendersToCountAtTillReconcile** parameter must include all the tenders to count for both base and alternate currencies during till reconciliation. For example:

Example 2–6 Parameters to support Krona as the base and USD as the alternate currency

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SOURCE PUBLIC "SOURCE"
"classpath://com/extendyourstore/foundation/tour/dtd/paramsourcescript.dtd">
<SOURCE name="register">
<GROUP hidden="N" name="Tender">
<PARAMETER final="N" hidden="N" name="StoreCreditsAccepted" type="LIST">
<VALIDATOR class="EnumeratedListValidator"
package="com.extendyourstore.foundation.manager.parameter">
<PROPERTY propname="member" propvalue="None"/>
<PROPERTY propname="member" propvalue="USD"/>
<PROPERTY PROPNAME="MEMBER" PROPVALUE="SEK"/>
<PROPERTY PROPNAME="MEMBER" PROPVALUE="EUR"/>
</VALIDATOR>
<VALUE value="SEK"/>
<VALUE value="USD"/>
<VALUE value="EUR"/>
</PARAMETER>
<PARAMETER final="N" hidden="N" name="ChecksAccepted" type="LIST">
<VALIDATOR class="EnumeratedListValidator"
package="com.extendyourstore.foundation.manager.parameter">
<PROPERTY propname="member" propvalue="None"/>
<PROPERTY propname="member" propvalue="USDCHK"/>
<PROPERTY propname="member" propvalue="SEKCHK"/>
<PROPERTY propname="member" propvalue="EURCHK"/>
</VALIDATOR>
<VALUE value="SEKCHK"/>
<VALUE value="USDCHK"/>
</PARAMETER>
<PARAMETER final="N" hidden="N" name="CashAccepted" type="LIST">
<VALIDATOR class="EnumeratedListValidator"
package="com.extendyourstore.foundation.manager.parameter">
<PROPERTY propname="member" propvalue="None"/>
<PROPERTY propname="member" propvalue="USD"/>
<PROPERTY propname="member" propvalue="SEK"/>
<PROPERTY propname="member" propvalue="EUR"/>
</VALIDATOR>
<VALUE value="SEK"/>
<VALUE value="USD"/>
</PARAMETER>
<PARAMETER final="N" hidden="N" name="GiftCertificatesAccepted" type="LIST">
<VALIDATOR class="EnumeratedListValidator"
package="com.extendyourstore.foundation.manager.parameter">
```

```
<PROPERTY propname="member" propvalue="None"/>
<PROPERTY propname="member" propvalue="USD"/>
<PROPERTY propname="member" propvalue="SEK"/>
<PROPERTY propname="member" propvalue="EUR"/>
</VALIDATOR>
<VALUE value="SEK"/>
</PARAMETER>
<PARAMETER final="N" hidden="N" name="TravelersChecksAccepted" type="LIST">
<VALIDATOR class="EnumeratedListValidator"
package="com.extendyourstore.foundation.manager.parameter">
<PROPERTY propname="member" propvalue="None"/>
<PROPERTY propname="member" propvalue="USDCHK"/>
<PROPERTY propname="member" propvalue="SEKCHK"/>
<PROPERTY propname="member" propvalue="EURCHK"/>
</VALIDATOR>
<VALUE value="SEKCHK"/>
<VALUE value="USDCHK"/>
Configuring a New Base Currency
Appendix: Changing and Configuring a New Base Currency D-7
</PARAMETER>
</GROUP>
<GROUP hidden="N" name="Reconciliation">
<PARAMETER final="N" hidden="N" name="TendersToCountAtTillReconcile" type="LIST">
<VALIDATOR class="EnumeratedListValidator"
package="com.extendyourstore.foundation.manager.parameter">
<PROPERTY propname="member" propvalue="Cash"/>
<PROPERTY propname="member" propvalue="Check"/>
<PROPERTY propname="member" propvalue="ECheck"/>
<PROPERTY propname="member" propvalue="Credit"/>
<PROPERTY propname="member" propvalue="Debit"/>
<PROPERTY propname="member" propvalue="TravelCheck"/>
<PROPERTY propname="member" propvalue="GiftCert"/>
<PROPERTY propname="member" propvalue="Coupon"/>
<PROPERTY propname="member" propvalue="GiftCard"/>
<PROPERTY propname="member" propvalue="StoreCredit"/>
<PROPERTY propname="member" propvalue="MallCert"/>
<PROPERTY propname="member" propvalue="PurchaseOrder"/>
<PROPERTY propname="member" propvalue="MoneyOrder"/>
<PROPERTY propname="member" propvalue="USDCash"/>
<PROPERTY propname="member" propvalue="USDTravelCheck"/>
<PROPERTY propname="member" propvalue="USDCheck"/>
<PROPERTY propname="member" propvalue="USDGiftCert"/>
<PROPERTY propname="member" propvalue="USDStoreCredit"/>
</VALIDATOR>
<VALUE value="Cash"/>
<VALUE value="Check"/>
<VALUE value="ECheck"/>
<VALUE value="Credit"/>
<VALUE value="Debit"/>
<VALUE value="TravelCheck"/>
<VALUE value="GiftCert"/>
<VALUE value="Coupon"/>
<VALUE value="GiftCard"/>
<VALUE value="StoreCredit"/>
<VALUE value="MallCert"/>
<VALUE value="PurchaseOrder"/>
<VALUE value="MoneyOrder"/>
<VALUE value="USDCash"/>
<VALUE value="USDTravelCheck"/>
<VALUE value="USDCheck"/>
```

```
<VALUE value="USDGiftCert"/>
<VALUE value="USDStoreCredit"/>
</PARAMETER>
</GROUP>
</SOURCE>
```

Resource Bundle Configuration

New resource bundle keys describing the new currency, its issuing country must be added to the Point-of-Service resource bundle commonText, ejournalText, tillText, dailyOperationsText, and parameterText. For example:

Example 2–7 New commonText Resource Bundle Keys

```
#
# Supported Nationalities
Common.SE_Nationality=Swedish

#
# Supported Currencies
Common.SEK=Swedish Krona

#
# Supported Checks
Common.SEKCHK=Swedish Krona

#
#  Tender Types
#
Common.SEKCash=SEK Cash
Common.SEKCheck=SEK Check
Common.SEKTravCheck=SEK Trav. Check
```

Example 2–8 New ejournalText Resource Bundle Keys

```
JournalEntry.SEK=SEK
```

Example 2–9 tillText Resource Bundle Keys

```
SelectTenderSpec.SelectSEK=SEK

Add example for dailyOperations Resource Bundle Keys:

FinancialTotalsSummaryEntrySpec.CURRCODE_SE=SEK

Add example for parameterText Resource Bundle Keys:
Common.SEKCash=SEK Cash
Common.SEK TravelCheck=SEK  Traveler's Check
Common.SEK Check=SEK  Check
Common.SEK GiftCert=SEK  Gift Certificate
Common.SEK StoreCredit=SEK  Store Credit
Common.SEKGiftCard=SEK Gift Card
```

3

Returns Authorization

The integration allows for Point-of-Service to collect positive ID during the return transactions, to form and send the Return Request messages to Returns Management, to interpret and present the Returns Management Return Response messages, and to form and send Final Result messages to Returns Management right before the return transaction is completed. Point-of-Service has support/flow for all Returns Management Return Response types. Point-of-Service has support/flow for accepting and managing Returns Management recommended tenders.

Returns Management provides the ability to deliver an accept/deny response for attempted refunds on line items of return transactions as well as non-receipted return attempts through standard XML messages. The retailer can configure enterprise-wide, down to store- and item-specific, receipted and non-receipted policies that are applied to line items on transactions occurring at a point of sale or point of return. The policy definition as well as accept/deny logic is contained within the enterprise and therefore is abstracted from the point of sale or return such that Returns Management can work with any point of sale or return application, including web or phone order systems. Returns Management provides the ability to count instances of behavior for customers and cashiers based on negativity activity and deny returns based on frequent suspicious activity. Included are inquiry screens to research an attempted refund or a particular score and its history.

Exception Flow

Communication with Returns Management is available only when Point-of-Service Server is in ONLINE mode. If Point-of-Service Server goes offline at any time during authorization or sending Final Result, the authorization request and final result information will be saved in Point-of-Service as offline return information, the message in E-Journal is logged and the offline return information will be sent to Returns Management when it is available.

Error Handling

Error handling is limited to logging errors during the return authorization. The exceptions such as IOException and invalidItem that occur during WSService communication are re-thrown as WSException, as well as logged for error tracking and resolution.

Logging

RPI uses Log4J for Logging. The following logging levels can be used:

- **Info**: For logging information messages.
- **Debug**: For logging all the debug messages.
- **Error**: For logging application errors.

However, the logging level can be configured with log4J.xml.

Glossary

Batch

A collection of data operations that are processed during import at one time. The size is determined by a configurable parameter.

Bundle

A collection of import files, one file per data type, stored as a compressed archive containing a manifest. It is expected that the retailer or implementation team is responsible for delivering to the Store the bundle along with manifest for all data feeds to the Store. MOM applications can package the bundle but do not provide delivery functions.

Corporate

Used interchangeably with *enterprise*. The enterprise environment of the retailer where enterprise applications are deployed. Oracle Retail Central Office is deployed in the enterprise.

Data Access Object (DAO)

A Java class that can retrieve and persist data to and from a data source. DAO is well-known JEE development pattern.

Data Distribution Infrastructure (DDI)

The infrastructure and application components that are responsible for distributing seed data from enterprise applications to Store applications, ODS at Corporate (or enterprise), and Store Database at the stores.

Data Transfer Object (DTO)

A class that contains data records from a received payload. The DTO's attributes are populated with the parsed data.

DIMP

Data Import

Incremental

There are two types of update operation, full incremental and delta incremental. Full incremental assumes that all the fields for a data type are supplied in the XML. A delta incremental import contains only the fields that are being changed.

ISP

In-Store-Processor

JEE/J2EE

Java Enterprise Edition (formerly Java 2 Enterprise Edition) is a set of APIs designed to support tier 1 type business models.

Java Database Connectivity (JDBC)

An API used to communicate with relational databases.

Kill And Fill

Kill And Fill refers to a data operation where all the existing data in a table is deleted (kill) and then replaced with new data (fill).

Limit (discount rule)

The maximum price allowed for a source or target to be part of a deal. Used most often when the source or target is a classification or department where many different priced items exist.

Manifest

A file within a bundle that lists the data files in the bundle and their interdependencies.

Minimum Data

Minimum Data is defined as the minimum set of data necessary to support the deployment of Stores applications only.

If the user attempts to select any function or log in, an error may occur in the application without Sample Data loaded. See Sample Data.

Operational Data Store (ODS)

The corporate data repository that services Oracle Retail Central Office.

ORBO

Oracle Retail Back Office

ORCO

Oracle Retail Central Office

ORLT

Oracle Retail Labels and Tags

ORPOS

Oracle Retail Point of Service

ORRM

Oracle Retail Returns Management

ORSIM

Oracle Retail Store Inventory Management

ReSA

Oracle Retail Sales Audit

RMS

Oracle Retail Merchandising System

RPM

Oracle Retail Price Management

RTLog

Retail Transaction Log

Sample Data

A set of data used to demonstrate application features.

Store Applications

Oracle Retail applications that run in the store environment. This includes:

- Oracle Retail Back Office
- Oracle Retail Point-of-Service
- Oracle Retail Strategic Store Solutions
- Oracle Retail Labels and Tags
- Oracle Retail Store Inventory Management
- Oracle Retail Central Office
- Oracle Retail Returns Management.

It must be noted that even though Oracle Retail Central Office runs in the corporate environment, it is classified as a store application.

Store Database (SDB)

The data repository for store applications.

Strategic Store Solutions

The Oracle Retail business unit that assumes responsibility for applications running in the Store environment.

Threshold (discount rule)

The minimum price allowed for a source or target to be part of a deal. Used most often when the source or target is a classification or department where many different priced items exist.