Infor Reporting Development Guide for the Base Data Store Models

Release 10.5.x

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

Publication Information

Release: Infor Reporting 10.5

Publication Date: April 28, 2016

Contents

About this guide

This guide provides information about the Framework Model in the Infor Business Vault Base Data Store. It explains the organization of the data from the Base Data Store and how to access and customize the data using IBM Cognos Framework Manager.

This guide includes these subjects:

- Modeling techniques and best practices
- The domain specific models
- The location of dimensions and transaction query subjects and how you can extend them

,

Intended audience

This guide is for technical users who are customizing the Base Data Store schema.

Note: This guide provide steps to help a Framework Manager modeler to change the existing models. Users must be familiar with the Framework Manager tool. This guide does not provide information on how to use Framework Manager.

Related documents

You can find these documents in the product documentation section of the Infor Xtreme Support portal:

- *Infor Business Vault Installation Guide*
- *Infor Business Vault Installation Guide for the Base Data Store*
- *Infor Reporting Installation Guide for the Platform*
- *Infor Reporting Installation Guide for the Base Data Store Reports*
- *Infor Reporting Installation Guide for the Base Data Store Models*
- *IBM Cognos Framework Manager User Guide*
- *IBM Cognos Framework Manager Guidelines for Modeling Metadata*

Latest information

For the latest information about this release, see Solution Record KB 1496547 on Infor Xtreme Support. This solution record may contain additional post-release installation information, known issues, and updates about this release. You should check this solution record periodically for the latest information.

Log on to Infor XtremeSupport at http://www.infor.com/inforxtreme. Specify **1496547** in the **Incident/KB/Documentation/Defect ID** field and click the arrow to retrieve this solution.

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal.

If we update this document after the product release, we will post the new version on this website. We recommend that you check this website periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

About the Framework Model

1

The Framework Model contains metadata that describes the data in one or more data sources. The metadata includes physical information and business information that drive query generation for Infor Reporting.

The IBM Cognos Framework Manager is a metadata modeling tool that you can use to customize the Framework Model. You can use this tool to organize the metadata to create different views of the data in the data source and provide different audiences a global representation of your business information. You can also apply security and multilingual capabilities. After you customize the model, you can create the package and deploy it to your end users for reporting.

In Framework Manager you work on a local version of the model. You must publish, or deploy, the model to the server to make it available to Infor Reporting.

Terminology

Term	Definition
Cardinality	A numerical indication of the relationship between two query subjects, query items, or other model objects.
Dimension	A broad grouping of descriptive data about a major aspect of a business, such as products, dates, or locations.
Model	A physical or business representation of the structure of the data from one or more data sources. A model describes data objects, structure, groupings, and relationships.
Namespace	A namespace contains a physical or a collection of physical tables. Each name must be unique within a namespace. For example, a namespace for the physical layer contains the physical tables that are displayed in the database. The namespace

Term	Definition
	for the business layer may have one query subject that represents a collection of related tables.
Package	A subset of a model made available to end users to create reports.
Project	A set of models, packages, and related information.
Query item	A representation of a column of data in a data source. Query items can display in a model or in a report. They contain a reference to a database column, a reference to another query item, or a calculation.
Query Subject	A named collection of query items that are closely functionally related.

Best practices, standards and techniques

Use these best practices and standards when you customize the Framework Model with Framework Manager.

Model durability

User requirements often change and can affect the model. The goal is to create a model that can survive these changes without affecting the end user and existing queries and reports. Renaming query items is one of the most frequent changes that modelers must implement in their models. If the model is durable, you can perform these changes with no effect on existing reports. In a multilingual environment a durable model can have language specific labels for query items without the risk of breaking existing reports.

Model layers

The model has three layers that help to ensure the durability of the model.

- Physical layer

  The physical layer contains the metadata imported from, or directly related to, the source database. The table, view, or column names from the database are referenced. All joins between tables are defined in this layer.
- Business layer

The business layer contains the metadata that is used to rename and reorganize the physical layer. The columns are given business names and are reorganized based on business needs.

- Presentation layer

  The presentation layer contains groupings of shortcuts to query subjects that are defined in the Business layer. The grouping of query subjects provides for ease of deployment and rendering in the authoring studios.

Framework model languages

To support multilingual metadata, you can add multiple languages to a model. Infor uses two locales for special purposes. English (Zimbabwe) is used as the active language and English (Botswana) is used as the design language.

- Design language

  When you create reports, you use the design language. The design language is used as a reference to the field, not the actual language. Therefore, if you change, for example, the English (US) or French value for a field, the report remains valid. The design language of the model is assigned when the model is created and it cannot be changed.

- Active language

  The active language is used to help make models more durable. When you add or update queries, the active language must be English (Zimbabwe). You must switch to English (Zimbabwe) before making any modifications.

  Select **Project > Define Languages** and select *English (Zimbabwe)* as the active language. You can rename query items in all other locales except English (Zimbabwe).

Joins

When you complete your joins for a new dimension or fact, you should generate a context diagram and confirm that no lines are crossed. Select the **Launch Context Explorer** to generate a context diagram and review the joins between a set of tables.

Customizing the model

When you customize the model, we recommend that you use these best practices:

- Relationships in the models should only exist at the physical view layer. They do not exist anywhere else in the model. Relationships can exist in other namespaces of a model but we recommend that you only have the relationships in the physical layer.
- The SQL that is used for most tables should be a `select * from xx=table name`.

- All queries for the model use the Max Variation Table to access the most recent transactions either through the generated SQL or an existing table.
- A dimension can only be joined to one transaction table at a time. To join to different transaction tables, create multiple instances for a dimension. Use the next dimension in a sequence to join the transaction table.

Cardinality

Fact to Fact (Header/Line)	1:n to 1:n
Fact to Supporting tables	1:n to 0:n
Fact to Notes tables	1:1 to 0:1
Dimensions to Fact	1:1 to 1:n
Dimensions to Max table	1:1 to 1:1
Dimensions to Note tables	1:1 to 0:1

Understanding instances of dimensions in a functional area model

2

In Infor Reporting, the master noun tables are in the physical layers of the functional area models. The master noun tables are organized within folder structures. There is a top folder for the name of the noun. Additionally, there are individual folders for each of the master noun instances that are required in the functional area model.

The first instance of the noun contains the actual physical tables. Every subsequent instance folder is a shortcut alias to the first instance. The technique of using shortcut aliases provides these advantages for modeling:

- You can reduce the size of the model because only the first instance contains the physical tables.
- You can update the physical table in instance one and all other instances can use the updates.

Ensure there is the same number of shortcuts and joins from instance to instance or folders to folders. Every instance should point to the same database table.

These are the model projects that are delivered with Infor Reporting:

Project Name (Models)	Package Name
I10R_A and D Manufacturing	Aerospace and Defense Manufacturing Reporting
I10R_Accounts Payable	Accounts Payable Reporting
I10R_Accounts Receivable	Accounts Receivable Reporting
I10R_BV_Metadata	Metadata Reporting Components
I10R_General Ledger	General Ledger Reporting
I10R_Purchasing	Purchasing Reporting
I10R_Sales	Sales Reporting
I10R_Sales Schedule	Sales Schedules
I10R_Warehouse	Warehouse Reporting

Adding dimensions

You can create a master table or an additional instance of a master table. For example, if a relationship for the CUSTOMER master noun must exists for another transaction table in the same model, the CUSTOMER noun must be brought into the physical layer multiple times with a different name to be used for joining other tables.

You can add the table to the functional model. You must add this information to the physical and business layers:

- Add the database table(s) to the physical layer of a model for example, CUSTOMER.
- Add a folder and place all of the columns in the query subject of the business layer namespace.
- Make the customer information available through a package for an end-user's consumption.

See "Creating a business transactional query" on page 21 to update the business layer.

These steps are to add new dimensions to the physical layer of a functional model, such as **I10R_Sales**:

1. Select **Project > Languages > Define Languages > Active Language** and ensure that the active language is set to **English (Zimbabwe)**.
2. To open the Sales model, select **I10R_Master** and expand **(en-zw) I10R_Sales**.
3. In the project viewer, expand **Data Sources**.
4. In the Properties pane, click the Catalog ellipses.
5. In the Edit Property Value pane, specify the name of the database where the dimension tables reside, for example, **BV_BDS_IB839**.
6. Click **OK**.
7. Right-click the **(en-zw) Physical View Sales** namespace and select **Run Metadata wizard**.
8. Click **Next** twice.
9. On the Select Connection pane, select the data source connection that you specified in step 5.
10. Click **OK**.
11. Select **Business Vault > BV_BDS_IB839** and expand **dbo** and **Tables**.
12. Select all the objects, (tables and views), to import to the new dimension.
13. Click **Next**.
14. Clear the **Use primary and foreign keys** check box.
15. Click **Import**.
16. For each query that you import into the model, right-click and select **Edit definition**.
17. To define the keys for the table, click **Determinants** on the Query Subject Definition properties pane.
18. Right-click the **Name**, select **Rename** and specify a determinant name.
19. Delete the determinant keys, and add them in the same order as they are in the SQL database.
20. Change the query item usage property to **attribute** for dates, indicators, and for any other field that does not require a total by the tool. To set the **Usage** property to **attribute** instead of identifier or fact, expand each query and select a field. To make the same change to several fields, press the Ctrl key and select multiple fields.
21. Change the **Usage** property to **Attribute** for the first field.

22 To change the **Usage** property to **Attribute** for all the remaining fields, select the drop down arrow to the right of the Usage value for the second field to change and drag it to the last row in the property pane.

Note: The **Usage** property for the key fields must remain as **Identifier** in all the tables.

23 To create relationships between tables, join from the primary table to the secondary table and join on all the keys from left to right. Select two tables to join, right-click and select **Create > Relationship**.

24 To join on all the keys in each table or query subject column, select the corresponding query items from left to right and click **New Link**. For example, for the join between the CU_CUSTOMER_NT table and the CU_CUSTOMER_NT_LC table, select the query item **N_PID_ID_VARIATION_ID** in each column and click **New Link**.

The cardinality value is defaulted to a 1:1 relationship for the main table and the secondary table. Confirm the cardinality and click **OK**.

See "Cardinality" on page 10.

25 Repeat this process for each set of tables that you want to join.

Adding a folder instance

You can add a folder instance to organize multiple instances of the same noun or to organize noun folders. To maintain the star schema, you must use folders to organize your nouns. Each transaction table must be joined to its corresponding noun instance.

See "Adding dimensions" on page 12.

1 After the noun tables are in the functional model, right-click the physical view namespace and select **Create folder**.

2 Specify a mixed case folder name. Prefix the name with (en-zw), for example, **(en-zw) Product**.

3 Click **Next**.

4 Click **Finish**.

5 Select the new folder. Right-click and select **Create folder**. Prefix the name with (en-zw), for example, **(en-zw) Product 1**.

6 Select all the product tables that are in the model. To select all the tables, press the control key and select each table.

7 Drag and drop the selected tables into the **(en-zw) PRODUCT1** folder.

See "Adding dimensions" on page 12.

8 Rename all of the tables in the **(en-zw) PRODUCT1** folder by placing a **1** after the main table name.

For example, specify PRODUCT1, PRODUCT1_ITEM_LOCATION, PRODUCT1_COMPONENTS. Specify the names for the English (Zimbabwe) and the English (United States) languages

9 If an additional instance of a noun is required, select **(en-zw) Product**. Right-click and select **Create a new folder**.

10 Specify **(en-zw) PRODUCT2** as the folder name. Click **Next** and **Finish**.

11 Expand the **(en-zw) PRODUCT1** folder and select all the tables in the folder.

12 For the tables that you selected, right-click and select **Create alias shortcut**.

13 Move the alias shortcut into the **(en-zw) PRODUCT2** folder.

14 Rename all alias shortcuts by prefixing the table names with **2** and removing **Shortcut to**.

For example, re-name Shortcut to PRODUCT1 to PRODUCT2 for English (Zimbabwe) and English (United States) languages.

15 Create the same relationships as in the Product 1 folder. If you receive the message `A relationship between these two query subjects already exists`, click **No** and manually create the relationship.

We recommend the use of shortcut aliases to reduce the size of the model. These are queries that point back to the original physical table(s). For example, the Item Master has multiple aliases created against the CODE table and these are used to join the classification codes. After the first relationship for these aliases exists, and you try to create the next relationship, the message is displayed because a relationship already exists between these tables. Click **No** because the next set of fields to join the alias is different then the first one.

16 Set the correct cardinality.

See "Cardinality" on page 10.

17 Click **Validate** to validate the relationship.

18 Repeat these steps for each additional instance of the noun.

Max variation processing

All queries in the domain specific models are joined to a max variation table. A max variation table retrieves the most recent transactions from each of the corresponding tables. Multiple variation IDs are sent to the Base Data Store and the max variation tables have the latest variation ID. This is used to establish a relationship between the max variation table and regular table. This relationship is used to retrieve the most recent master or transaction record to present to the end-user.

When you add customer specific tables to the Infor Base Data Store, you must also add a corresponding max variation (MV) table. A max variation table is required for master noun tables and transaction tables. You can add a max variation table or you can use an SQL statement to create a query subject in the functional model physical view. Join the query subject to the new table to retrieve the most recent transactions.

For example, this SQL statement retrieves the most recent Customer from the Customer table:

```
Select PID_NID,MAX(PID_ID_VARIATION_ID) as MAX_ PID_ID_VARIATION_ID
From [Business Vault]. CU_CUSTOMER
GROUP BY PID_NID
```

Linking the max variation table

1 Create the max variation table or the query subject in the model.

See "Adding dimensions" on page 12.

2 Specify **Identifier** for the **Usage** properties for the `PID_NID` and `PID_ID_VARIATION_ID` on the query subject, for example, **CU_CUSTOMER1_MAX_VARIATION**.

3 Right-click the query subject, for example **CU_CUSTOMER1_MAX_VARIATION**, and select **Edit > Definition**.

4 To define the keys for the table, click the **Determinants** tab in the Query Subject Definition properties pane.

5 Right-click **Name**, select **Rename** and specify a new determinant name.

6 Delete the determinant keys and add them back in the correct order. The keys must be listed in the order in which they appear in the SQL database.

7 To join the two tables, select the max variation query subject and the corresponding table. For example, select `CU_CUSTOMER1` and `CU_CUSTOMER1_MAX_VARIATION`, right click and select **Create > Relationship**.

8 Join the two tables on the keys and add **S_CD NOT IN ('Deleted','Pending')** as part of the join to exclude records with a Deleted or Pending status.

For example, specify this code:

```
CUSTOMER1.CU_CUSTOMER1_MAX_VARIATION.PID_NID = CUSTOMER1.
CU_CUSTOMER1.PID_NID
 AND CUSTOMER1. CU_CUSTOMER1_MAX_VARIATION.PID_ID_VARIATION_ID =
CUSTOMER1. CU_CUSTOMER1.
PID_ID_VARIATION_ID AND CUSTOMER1.CU_CUSTOMER1.S_CD NOT IN
('Deleted','Pending')
```

9 Set the correct cardinality between the two tables.

10 Click **OK**.

Saving your model changes as a script

You can add or update the base models in the Infor Base Data Store. You can add new queries and join them to any of the existing models. This does not affect the original queries. You can add a new query subject that contains new fields to append to an existing table. The key attributes of a query subject must be created in the model to join to its original dimension. Any modifications that you make to the base models must be tracked. You cannot modify the delivered original query subject.

All your changes are logged by the system and these logs can be saved as a script. We recommend that you save your changes to the models as a script. When you receive a new version of the models, you can re-apply your changes by running the scripts. All the steps must be saved as part of the script.

You can use the scripts to reapply these types of changes:

- New table additions
- New relationships
- Adding or renaming columns

These modifications can only be made to the original queries in the Base Data Store:

- Rename a query item in the Business View
- Add filters

Additionally, Infor may provide scripts to update various models. If a new model is provided in a future release, you can re-run your script and retain your changes.

To save your changes as a script:

1. Select the model that has your changes. Click **Project > View Transaction History**.
2. To display the log files by date and user, expand the first transaction list.
3. Locate the log file that contains your changes.
4. Expand the log file and select all the transactions that contain your modifications.
5. Select the **Save as script** tab. Save the script on the server where the models reside.

Organizing and creating business queries 3

After you make your changes in the physical layer, you can organize and rename the noun and transaction elements into folders in the Business View before you make it available to your end-users. For example, you can change field names from their actual names to user-friendly names in a particular business area. We recommend these standards for organizing your business queries.

In some business cases, you may require two separate queries in the model to support two distinct business scenarios. One package can have two versions of the same query that are presented to the end-user. The first scenario includes a set of reports that provide a view of the most recent transactions and queries. These reports are filtered using the Max Variation ID, Max revision ID, or a combination of the two. The second set of reports does not have the Max Revision ID filter applied and provides the entire history of the business.

In the package, the two queries are organized into separate namespaces for ease of use. The content, layout, and look and feel, of the namespaces are the same. Reports can be built from one namespace at a time.

Copy dimension elements of a particular noun or transaction into the header or line business queries from the functional model physical layer. Note the instances and the tables to which the instances are joined. We recommend that you create at least one header folder and a folder for each of the dimensions that belong to the business query. Move the transaction elements into the transaction folder and move the first dimension elements individually into the correct dimension folder. Repeat the process for the remaining dimensions for the header and the line.

After you complete this task, create additional subfolders under the main folders to organize the information. Ensure you are working in the English (Zimbabwe) language and using mix cases when renaming. Use the Base Data Store queries as examples to structure your business queries in a similar way.

Transactional query folder structure

All the elements and a folder for all of the Master Data tables to which they are joined, are contained in a transactional business query. This information is organized in folders and subfolders for ease of use and navigation. The number of folders and subfolders varies for transactional nouns depending on the number of sub-tables it contains.

Folder structure process

1. Create an individual business transaction query, for example, Order Header. This is a representative of a header, line, or sub-table.
2. Move all of the columns (query elements) into each corresponding transaction query.
3. Organize, re-order, re-name, add calculations, and filters for these elements inside the folders.
4. Any master noun that is joined to one of these tables will have a folder representative of all the elements that pertain to this noun. These nouns elements should be organized in the same fashion as the transaction elements for ease of use by the end-user. There are folders within folders to further organize the elements.

Folder naming and organizing guidelines

- Folders are in alphabetical order. The exception is the first folder that is the transaction folder and the last 3 folders which are always Customization Details, Keys, and Audit.
- The transaction query subjects are ordered in the normal business transaction flow. For example, the order for Sales is contract, sales order, shipment, invoice, return. For Purchasing it is requisitions, RFQs, contracts, purchase orders, ASNs, deliveries, and invoices.
- There are some hidden folders such as Company Mailing Address, ERP-Specific Classification Codes, ERP-Specific User Fields, Secondary Contacts, and Secondary Mailing Address. These folders are in the model but are not visible to end users when a package is published.
- All query items are in alpha order except the first transaction query subject folder and the Master Data Noun Details folder. These folders are in alpha order except for the initial query items which are in this order:
  - ID
  - Drillback
  - Description
  - Display ID
  - ID-Name
  - Name
- List IDs and Locale query items are always hidden.
- Currency query items, such as transaction, base, and reporting, should be hidden. Tax authority should be displayed.
- UOM Base should be hidden and UOM Transaction should be displayed.
- The transaction query subject header details folder should contain these three calculated columns. If there are no monetary elements on the header, then the Currency Code can be on the line.
  - Currency Code
  - Currency Code (Base)
  - Currency Code (Report)

- UOM elements are on the transaction query subject line details folder and should contain two calculations for the transaction and base unit of measure.
  - UOM
  - UOM (Base)
- Query items in the transaction query subject Contacts folder should be hidden if there is a duplicate in the Master Data Contacts folder.
- You should complete moving and structuring all of the transaction query elements before you add the required Master Data elements.
- All Customization Details folders contain User Fields. A Classification Code folder is present when the specific BOD contains classification code elements. The order is alphabetical for folders and query items.
- The Keys folder for the transaction query subject should contain NIDs for associated Master Data tables and Transaction tables associated with the specific domain. For example, the Sales Domain should contain Customer NID, Remit-to Location NID, Contract NID, Sales Order NID, etc. It should not contain Project NID, PO NID, Receivable NID because they are not in the specific domain. The transaction details folder contains the ID.
- The Keys folder for the transaction query subject header should contain the Max Variation ID, for example, Order Max Variation ID. Additionally, it should contain the Source (LID), for example, Order Source. The elements in the Keys folder are in alphabetical order.
- Bill-to Party, Company, Customer, Pay-from Party, Remit-to Location, Salesperson, Ship-from Location, Product, and Ship-to Party are all Master Data tables that are accessed by the transaction query subject (Contract). If you look at the Contract SQL Server tables, there are NIDS in the table.

Order Header folder structure example

This is an example of the Order Header folder structure:

- Order Header
  - Order Header
  - Bill-to Party
  - Company
    - Company Details
      - Company ID
      - Company ID-Name
      - Company Display ID
      - Company Desc
      - Company Status
      - Company Starus Date
      - Company Status Datetime
    - Company Additional Details

- Company Mailing Address
- Contacts
- Customer
- Distributed Changes

Folder structure for a transactional query

This is an example of the folder structure for a transactional query:

```
(en-zw) xx=Transactional noun top folder (example: (en-zw) Order Header)
(en-zw) Contacts
(en-zw) zz=Master Data folders (example: (en-zw) Bill-to Party, (en-zw)
 Customer)
(en-zw) yy=Supporting tables folders (example: (en-zw) Distributed
Charge)
(en-zw) Notes
(en-zw) Customization Details
   (en-zw) Classification Codes
   (en-zw) User Fields
       (en-zw) Amounts
       (en-zw) Date Time
       (en-zw) Dates
       (en-zw) Flags
       (en-zw) Integers
       (en-zw) Numeric
       (en-zw) Quantity
       (en-zw) String
       (en-zw) Time
(en-zw) Keys
(en-zw) Audit
```

Business query filters

There are two required filters that you must add to all business queries that you create. The Max Variation filter ensures that you always select the most recent information. The Status filter is used to select only the appropriate statuses.

(en-zw) Max Variation ID

Add one of these filters to your business queries:

```
([(en-zw) Physical View
Sales].[SALES_ORDER_CUSTOMIZED].[SOH_DID_ID_VARIATION_ID] =[(en-zw)
Physical View Sales].[
SALES_ORDER_CUSTOMIZED_MV].[SOH_DID_ID_VARIATION_ID]
```

OR

```
[(en-zw) Physical View Sales].[
SALES_ORDER_CUSTOMIZED].[SOH_DID_NID] IS NULL)
```

(en-zw) Status

Add one of these filters to your business queries:

```
([(en-zw) Physical View Sales].[
SALES_ORDER_CUSTOMIZED].[SOH_S_CD] not in (
'Deleted','Pending','Complete','Working','Modified')
```

OR

```
[(en-zw) Physical View Sales].[
SALES_ORDER_CUSTOMIZED].[SOH_DID_NID] IS NULL)
```

Creating a business transactional query

After you make your changes in the physical layer, you can organize the information in the Business View. Then you can make it available to your end-users. If you create a new table in the physical layer, you must add the query subject to the Business View.

1 Select **Project > Languages > Define Languages > Active Language** to ensure that **English (Zimbabwe)** is set as the active language.

2 To open the model, select **I10R_Master** and expand a project, for example, **(en-zw) I10R_Sales**.

3 Right-click **(en-zw) Business View** and select **Create > Query Subject**.

4 Specify the new query subject name and prefix it with **(en-zw)**. Select **Model (Query Subjects and Query Items)**.

5. Click **OK**.
6. On the Available Model Objects screen, expand **(en-zw) Physical View** and locate the table or tables to include in the transaction query.
7. To move the query elements, select a table. Select the elements in the table and drag into the Query items and Calculations pane on the right.
8. Click **OK**.
9. Click the **Filters** tab and click **Add**. There are two required filters that you must add to all business queries that you create, Max Variation ID and Status.
10. Specify **(en-zw) Max Variation ID** in the name field for the filter.

    See "Business query filters" on page 20.
11. Expand the Physical View and locate the Primary table. Use the variation ID fields with the variation ID from the max variation (MV) table to create an expression, for example:

```
 [(en-zw) Physical View].[ACNTING_ENTITY].[ID_ID_VARIATION_ID]=[(en-zw)

 Physical View].[ACNTING_ENTITY_MV].[ID_ID_VARIATION_ID]
```

12. To exclude the Deleted and Pending statuses, click the **Filters** tab and click **Add**.
13. Specify **(en-zw) Status** in the name box for the filter and define the filter. For example:

```
 [(en-zw) Physical View].[ACNTING_ENTITY].[S_CD] not in
('Deleted','Pending')
```

14. Click **OK**.
15. Create your folder structure.

    See "Transactional query folder structure" on page 17.
16. Rename each date field. Add **TIME** to the end of each column name.
17. To add a new calculated query item that includes the original date column, right-click on the query subject and select **Create > Query Item**. Specify the same name as the original date column.
18. In the expression pane, cast the original date as date, for example:

```
cast([(en-zw) Physical View
Sales].[SALES_ORDER].[SOH_CNCL_DLVRY_BY_DATE] as date)
```

19. Repeat steps 16 through 18 for all your date fields.
20. Change your active language from English (Zimbabwe) to English (United States). Define a business name for all the business query elements that you created. Define a business name, using mixed-cases, that your users can understand. Do not change the English (Zimbabwe) name.

    Note: You may be required to re-name query items to prevent duplicate names. Query items must be unique within the query. For example, the NID in the key Customer Master Data folder within the transaction query contains Party as part of the name, Customer Party NID. The NID in the transaction folder is Customer NID.

21 Organize your query elements into each folder and subfolder for ease of use and navigation.

Master data elements and folder structure in a transactional query

4

There are numerous query items for a master noun and for a transaction table. For ease of use and navigation, you should organize the query items into multiple folders. Because the Business Layer namespace is used by users, you should create business queries and folder structures that facilitate navigation of the Business Layer to build reports.

For example, create a business query and specify the name **(en-zw) Shipment Header**. Within the Shipment Header query, create additional folders. Put the most widely used query elements in the first folder. Continue to create additional folders to organize remaining transaction query elements until a desired folder structure is achieved.

The Shipment Header business query also includes master noun information such as (en-zw) Customer. Create a customer folder and within the customer folder create additional folders to structure the information in a similar way as the transaction elements.

Guidelines

These are techniques for naming and organizing your folders:

- The Keys folder for the Master Data should contain only the NIDs for sub-tables like Components, Notes, and EDI and Parents. For example, in the Customer Keys folder there is the Account NID, EDI NID, and Note NID. The Customer NID is not in the folder. The Customer NID is in the Keys folder for the transaction query subject. The Customer NID should be hidden in the Master Data Keys folder and it should be named Customer Party NID to prevent duplication with the Transaction Query Subject Keys folder.
- The Keys folder for the Master Data query items should begin with the master data name, for example, Bill-to Company and Bill-to Source. In the Location folders like Remit-to Location, the Keys folder Location query item should be titled Remit-to Security Location. This is required because the transaction area contains the same business view name Remit-to Location.
- The same master table can be used across all of the models and renamed depending on the domain. For example, when the Item Master is displayed in Sales, the folder is called Product and the query elements include product information such as, Product Class. In Purchasing, the folder is called Item and the query elements include item information such as, Item Class. The one exception is

Product Line which only applies to Sales and is hidden in Purchasing. The other element that is always hidden is Item/Product Status Desc that is NULL.

- When new columns are added to any of the Infor Base Data Store tables and the changes affect any of the models, we recommend that you save these changes as a script. You can run this script against your models.

  See "Saving your model changes as a script" on page 15.
- If you must make modifications that are specific to your environment, you should add a new table to the model that you can join to an existing Master Data or transaction table.
- If you rename columns or add new calculations, we recommend that you use the **Project** tab in Framework Manager and view transaction history to save your changes as a script. If a new model is provided, you can re-apply your saved modifications scripts.
- Do not modify the English (Zimbabwe) column. To do so would corrupt your reports.

Customer Master data folder structure example

This example shows a transaction query that has a Customer Master Data folder structure:

```
(en-zw) xx=Top Master Data folder (example: (en-zw) Customer, (en-zw)
 Bill-to Party)
(en-zw) Customer Details
(en-zw) Customer Contacts
(en-zw) Customer Mailing Address
(en-zw) Customer Additional Details
   (en-zw) Customer ERP-Specific Classification Codes
   (en-zw) Customer Defaults
   (en-zw) Customer EDI Processing
   (en-zw) Customer Notes
   (en-zw) Customer Secondary Contacts
   (en-zw) Customer Secondary Mailing Address
   (en-zw) Customer Customization Details
       (en-zw) Customer Classification Codes
       (en-zw) Customer User Fields
          (en-zw) Customer Amounts
          (en-zw) Customer Date Time
          (en-zw) Customer Dates
          (en-zw) Customer Flags
          (en-zw) Customer Integers
          (en-zw) Customer Numeric
          (en-zw) Customer Quantity
          (en-zw) Customer String
          (en-zw) Customer Time
   (en-zw) Customer Keys
   (en-zw) Customer Audit
```

Adding Master Data elements to a transactional query

1. Right-click the transaction query subject and select **Create > Query Item Folder** to create the main Master Data folder.
2. Re-name the folder for the first Master Data that you must add, for example, **(en-zw) Customer**.
3. Add the folders and sub-folders to the new Master Data folder. Re-name these folders for all available project locales.
4. Right-click the transaction query and select **Edit definition**.
5. On the Available Model Objects screen, expand **(en-zw) Physical View** of the functional model and locate the tables of the instance to include in the transaction query.

   Note: To avoid cross joins, ensure that the instance for which you are inserting elements, is the instance that you joined to the transaction query.

6. Change your active language from English (Zimbabwe) to English (United States) and re-name all the Master Data query items.
7. Rename each date field. Add **TIME** to the end of each column name.
8. Right-click the query subject and select **Create > Query Item**. Specify the same name as the original date column.
9. In the expression pane, cast the original date as date, for example:

```
CAST([Physical View Sales].[CUSTOMER1].[S_EFF_DATE] AS DATE)
```

10. Click **OK**.

Adding transactional query shortcuts to the presentation layer

You must create shortcuts to the Business View to create the presentation layer for your end users. Only the transaction query subjects that are associated with a specific domain are displayed on the first level, and the filters and prompt selections.

1. Hold down **Ctrl** and select all the query subjects and namespaces that you want to see when building reports. Select **Create > Shortcuts**.
2. While the query subjects and namespaces are selected, select **Edit > Cut**.
3. Select the **(en-zw) Presentation View** of your project. Right-click and select **Edit > Paste**.
4. Select English (United States) as the active language and rename the shortcuts that have been moved to the physical view. You must rename the shortcuts for every language in your presentation layer. See this example:

Shortcut	Rename
Shortcut to (en-zw) Order Header	Order Header

Shortcut	Rename
Shortcut to (en-zw) Order Lines	Order Lines

Creating the end user package

5

After you complete all your changes, you must make the package available to your end users for report writing.

To create an end user package:

1 In Framework Manager, right-click **Packages**.

2 Specify the package name in the Base Data Store:

Package name	**Description**
IR_MP	A and D Manufacturing Reporting
IR_AP	Accounts Payable Reporting
IR_AR	Accounts Receivable Reporting
IR_GL	General Ledger Reporting
IR_PM	Purchasing Reporting
IR_SA	Sales Reporting
SRM_SA	Sales Schedules
IR_WH	Warehouse Reporting

3 Click **Next**.

4 Expand the namespace, for example, Sales Reporting Components namespace.

5 Double-click the physical view of your project.

6 To exclude the physical view from the presentation, click the Business View namespace once. Your other namespaces should have a check.

7 Click **Next**.

8 Select the language to include as part of the package. The package may be in multiple languages.

9 Click **Next**.

10 Optionally exclude or include your database-specific functions.

11 Click **Finish**.

12 Click **Yes** to publish your package.