iWay Software
An Information Builders Company

iWay

Omni-Payer™ HealthViews User's Guide

Version 1.6.1

DN3502229.1216

Contents

Preface

This documentation provides prerequisites and instructions to configure Omni-Payer™ HealthViews.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Configuring Omni-Payer™ HealthViews	Provides an overview for Omni-Payer™ HealthViews.

Documentation Conventions

The following table lists and describes the documentation conventions that are used in this manual.

Convention	Description

`THIS TYPEFACE`
or
`this typeface`
	Denotes syntax that you must type exactly as shown.

this typeface	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
underscore	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{}	Indicates two or three choices. Type one of them, not the braces.

Convention	Description
|	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).
. . .	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at *http://documentation.informationbuilders.com*. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing every tips and techniques. Access Focal Point at *http://forums.informationbuilders.com/eve/forums*.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, *http://www.informationbuilders.com*. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of *www.informationbuilders.com* also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 A.M. and 8:00 P.M. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities. Be prepared to provide your six-digit site code (*xxxx.xx*) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following table lists the environment information that our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists the deployment information that our consultants require.

Adapter Deployment	
Container	
Version	
Enterprise Information System (EIS) - if any	
EIS Release Level	
EIS Service Pack	
EIS Platform	

The following table lists iWay-related information needed by our consultants.

iWay Adapter	
iWay Release Level	
iWay Patch	

The following table lists additional questions to help us serve you better.

Request/Question	**Error/Problem Details or Information**
Did the problem arise through a service or event?	
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error messages.	
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	

The following is a list of error and problem files that might be applicable.

❑ Input documents (XML instance, XML schema, non-XML documents)

❑ Transformation files

❑ Error screen shots

❑ Error output files

❑ Trace files

❑ Service Manager package to reproduce problem

❑ Custom functions and agents in use

- Diagnostic Zip
- Transaction log

For information on tracing, see the *iWay Service Manager User's Guide*.

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, *http://documentation.informationbuilders.com/connections.asp*.

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website, *http://education.informationbuilders.com*, or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our website, *http://www.informationbuilders.com/support*.

1 Configuring Omni-Payer™ HealthViews

Omni-Payer™ HealthViews is a set of scripts that implements de-normalized tables and views.

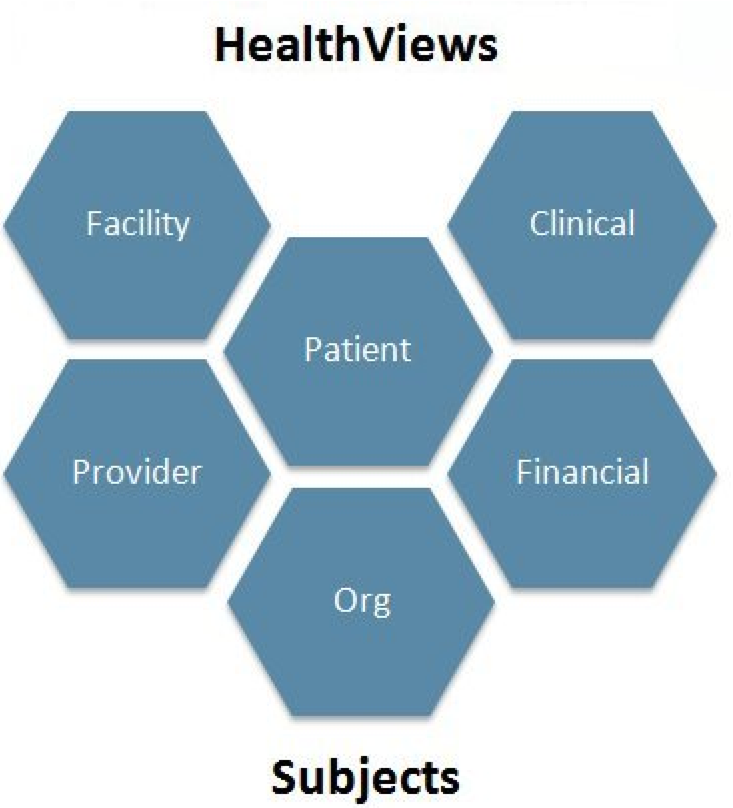
Omni-Payer™ HealthViews allows organizations to leverage existing clinical data, generating valuable actionable insights, which lead to tangible business results. Leveraging Omni-Payer™ HealthViews to compile and relate content across the entire organizational spectrum, business users can be empowered to communicate, visualize, and analyze data effectively.

This section provides an overview for Omni-Payer™ HealthViews, key features, and describes the configuration steps that are required.

Topics:

Overview

Omni-Payer™ HealthViews offers a commercial, off the shelf, clinical and operational data model with dynamic views to empower healthcare analytics and reporting.



Omni-Payer™ HealthViews provides a centralized data model, which is organized by Domain/Subject. In addition, client expansion and customization are supported.

Key features include:

- A centralized model to collect disparate healthcare data.
- Enables self-service healthcare business intelligence.
- A framework for developing clinical-centric analytic and data mining applications.
- Supports analysis for: Patient Movement, Demographics, Habits, Outcomes, Volumes, Infections, and more.
- A clinical health care COTS (commercial off the shelf) data model.

- The ability to be deployed as a standalone data model (RDBMS, MPP), or integrated with Omni-Patient™.
- Support for multi-tenant data storage, allowing for custom content to be stored and integrated with the Omni-Payer™ HealthViews data model.
- Data model relationships clearly defined to expedite content creation (reports, analytics).
- An optimized data model ensures rapid answers to difficult questions.
- The capability of Omni-Payer™ HealthViews to host data that generates Healthcare Performance Analytics dashboards, and balance scorecards.

Understanding the Architecture of Omni-Payer™ HealthViews

Omni-Payer™ HealthViews implements a star schema model that has been de-normalized for ease of use for reporting purposes. It is implemented as dynamic views over tables and provides the ability to view patient facts by date/time and also correlated clinical event data.

Key components include:

- **Healthcare data model (star schema).** Definition of common healthcare data and relationship for storage and reporting.
- **Dynamic database views of the data model.** A layer of abstraction of the data model to simplify business intelligence reporting.

Prerequisites and Supported Platforms

Omni-Payer™ HealthViews is currently supported on DB2 Version 10.5 and must be used only with production versions of Omni-Payer™.

Creating the Omni-Payer™ HealthViews Schema for DB2

If the Omni-Payer™ HealthViews schema does not already exist, then it must be created. To create the Omni-Payer™ HealthViews schema, run the following script:

```
0000_omnipayer137_DB2_create_schema_script.sql
```

When running this script, set the name to be used for Omni-Payer™ HealthViews. The default value is:

```
HEALTHVIEWS
```

Installing and Configuring the Hash64 Function

In this section:

Installing the Hash64 Function

Omni-Payer™ HealthViews relies on a Hash64 function to create unique keys for the tables in the Omni-Payer™ HealthViews scheme. In addition, the Hash64 function is used to create integer values for fields that are character strings in Omni-Payer™. Joining integer fields provides improved performance than joining character strings.

Installing the Hash64 Function

The Hash64 function code is contained in the *ibi_functions.so* file. A system administrator who has the required authorization, must place the *ibi_functions.so* file in a known location (the default is */udbhome/udbinst4/sqllib/function*) and set permissions to *755* (readable to all, executable by all).

Once there it is created by entering the following SQL command:

```
CREATE FUNCTION hash64(varchar(255))
RETURNS BIGINT
EXTERNAL name '/udbhome/udbinst4/sqllib/function/ibi_functions!hash64'
LANGUAGE C
PARAMETER STYLE SQL
DETERMINISTIC
NOT FENCED
NULL CALL
NO SQL
NO EXTERNAL ACTION
```

Note: Only a system administrator who has the required authorization must perform the steps described in this section.

Creating the Stored Procedure

Omni-Payer™ HealthViews uses two stored procedures called *drop_if_exists* and *getsubcode*, which must be created in the default schema.

To create these stored procedures, run the *drop_if_exists.sql* script and *getsubcode.sql* script, which are both located in the \stored_procs directory.

Executing the Scripts

In this section:

A KornShell (ksh) script has been created to execute all of the various Omni-Payer™ HealthViews scripts in the correct order.

This script relies on configuration settings that must be defined in the *db2.config* configuration file. The script must also be run in the *no hangup* (*nohup*) mode so that it can be run uninterrupted in the background. Two scripts provided (one to run all Omni-Payer™ HealthViews scripts automatically and one to run an individual script).

Configuration File Settings

This section describes the contents and settings of the *db2.config* configuration file.

❑ **database.type=db2**

Defines the database type, which must always be set to *db2*.

❑ **database.host=omnidb2qadns**

Defines the address of the database, which can be logical or an IP address.

❑ **database.name=omnidbqa**

Defines the name of the database.

❑ **database.username=cdeopdbd**

Defines the user name to log on to the database.

❑ **database.userpass=Blu3cdev**

Defines the password for the user of the database.

❑ **database.src_schema=omnipayer**

Defines the name of the schema for Omni-Payer™, which is usually set to *omnipayer*.

❑ **database.targ_schema=healthviews**

Defines the name of the schema for Omni-Payer™ HealthViews, which is usually set to *healthviews*.

- **database.method_schema=cdeopdbd**

  Defines the method for the schema.

Using the build_all.sh Script

In general, the *build_all.sh* script does not need to be edited. The first section defines variables, including the path to the *db2_config.properties* file. The default path is *$HOME/db_config/db2_config.properties*. Change this path only if you placed your *db2_config.properties* file in a different location.

Before you run the *build_all.sh* script, you must set the environment variables for DB2 by typing:

For the DEV environment:

```
. /udbhome/udbinst3/sqllib/db2profile
```

or

For the QA environment:

```
. /udbhome/udbinst4/sqllib/db2profile
```

You can run the *build_all.sh* script by typing:

For the DEV environment:

```
nohup sh build_all.sh > hv_dev.out 2>&1 &
```

or

For the QA environment:

```
nohup sh build_all.sh > hv_qa.out 2>&1 &
```

As mentioned earlier in this document, the *nohup* command instructs the *build_all.sh* script to run in the *no hangup* mode.

The >*hv_dev.out* or >*hv_qa.out* commands direct the output of the scripts to a log file called *hv_dev.out* or *hv_qa.out*.

The ampersand character (&) in the command string instructs the *build_all.sh* script to run in the background.

Full Load and Incremental Scripts

After loading the initial data from Omni-Payer to HealthViews, incremental scripts can be used. The incremental scripts only update the HealthViews tables with new records from Omni-Payer rather than reloading all of the data from scratch.

A separate directory below the full load scripts directory is provided for running incremental loads. Run the *build_all.sh* script from this location to execute an incremental load.

Note: The *build_all.sh* script contains the *L_OMD* variable. This variable is used to define the *as of* date. Typically this *as of* date is two days behind the current date. This variable must be set each time an incremental update is executed.

List of Available Scripts

The following table lists and describes all of the available scripts for Omni-Payer™ HealthViews.

Script	**Description**
000_Healthviews_DB2_create_schema_script.sql	Used to create the HealthViews schema, not normally run.
001_t_date_dimension.sql	Defines all dates in the system.
002_t_time_dimension.sql	Defines all times in the system.
003_t_dim_codeset.sql	Defines all codesets in the system.
003_t_dim_codeset_inc.sql	Incremental update of codesets in the system.
003_v_dim_codeset_lookup.sql	View for looking up codesets.
003_t_source_code_relation.sql	Describes all source code relations in the system.
003_t_source_code_relation_inc.sql	Incremental update of source code relations.
003_v_source_code_relation.sql	View for looking up source code relations.
300_t_member_m.sql	All mastered members in the system.
300_t_member_m_inc.sql	Incremental update of mastered members.
300_v_member_m.sql	View for mastered members.
302_t_member_identifier_m.sql	All mastered member identifiers in the system.
302_t_member_identifier_m_inc.sql	Incremental update of member identifiers.
302_v_member_identifier_m.sql	View for all mastered member identifiers.

Script	Description
303_t_member.sql	All member instances in the system.
303_t_member_inc.sql	Incremental update of all member instances.
303_v_member.sql	View for all member instances.
304_t_member_identifier.sql	All member identifiers in the system.
304_t_member_identifier_inc.sql	Incremental update of member identifiers.
304_v_member_identifier.sql	View for all member identifiers.
305_t_provider.sql	All providers in the system.
305_t_provider_inc.sql	Incremental update of providers.
305_v_provider_.sql	View for all providers.
306_1_tmp_events_create.sql	Temporary table used for various events.
306_t_encounter.sql	All encounters in the system.
306_t_encounter_inc.sql	Incremental update of encounters.
306_v_encounter.sql	View for all encounters.
307_t_procedure_event.sql	All procedure events in the system.
307_t_procedure_event_inc.sql	Incremental update for procedure events.
307_v_procedure_event.sql	View for all procedure events.
308_t_observation_event.sql	All observation events in the system.
308_t_observation_event_inc.sql	Incremental update for observation events.
308_v_observation_event.sql	View for all observation events.
309_t_diagnosis_event.sql	All diagnosis events in the system.
309_t_diagnosis_event_inc.sql	Incremental update for diagnosis events.
309_v_diagnosis_event.sql	View for all diagnosis events.

Script	Description
310_1_tmp_events_drop.sql	Drops the temporary table used for events.
310_t_payer_claim.sql	All payer claims in the system.
310_t_payer_claim_inc.sql	Incremental update for payer claims.
310_v_payer_claim.sql	View for all payer claims.
311_t_payer_claim_line_item.sql	All payer claim line items in the system.
311_t_payer_claim_line_item_inc.sql	Incremental update for payer line items.
311_v_payer_claim_line_item.sql	View for all payer claim line items.
312_t_payer_claim_line_item_diagnosis.sql	All payer claim line item diagnosis records in the system.
312_v_payer_claim_line_item_diagnosis.sql	View for all payer claim line item diagnosis records.
313_t_event_note.sql	All event notes in the system.
313_t_event_note_inc.sql	Incremental update of event notes.
313_v_event_note.sql	View for all event notes.
314_t_reference_range.sql	All reference range records in the system.
314_t_reference_range_inc.sql	Incremental update of reference range records.
314_v_reference_range.sql	View for all reference range records.
315_t_allergy.sql	All allergy records in the system.
315_t_allergy_inc.sql	Incremental update of allergy records.
315_v_allergy.sql	View for all allergy records.
316_t_pharmacy_prescription_order.sql	All pharmacy prescription orders in the system.
316_t_pharmacy_prescription_order_event_inc.sql	Incremental update of pharmacy prescription order events.

Script	Description
316_v_pharmacy_prescription_order.sql	View for all pharmacy prescription orders.
317_t_pharmacy_administration_route.sql	All pharmacy administration route records in the system.
317_t_pharmacy_administration_route_inc.sql	Incremental update for pharmacy administration route records.
317_v_pharmacy_administration_route.sql	View for all pharmacy administration route records.
318_t_provider_m.sql	All mastered providers in the system.
318_t_provider_m_inc.sql	Incremental update of mastered providers.
318_v_provider_m.sql	View for all mastered providers.
319_t_provider_identifier.sql	All provider identifiers in the system.
319_t_provider_identifier_inc.sql	Incremental update of provider identifiers.
319_v_provider_identifier.sql	View of all provider identifiers.
320_t_provider_identifier_m.sql	All mastered provider identifiers in the system.
320_t_provider_identifier_m_inc.sql	Incremental update of all provider identifiers.
320_v_provider_identifier_m.sql	View of all mastered provider identifiers.
321_t_family_history.sql	All family history records in the system.
321_v_family_history.sql	View for all family history records.
322_t_pharmacy_dispense_event.sql	All pharmacy dispense events in the system.
322_v_pharmacy_dispense_event.sql	View for all pharmacy dispense events.
323_t_medication_admin_event.sql	All medication administration events in the system.

Script	Description
323_v_medication_admin_event.sql	View for all medication administration events.
324_t_vaccination_admin_event.sql	All vaccination administration events in the system.
324_v_vaccination_admin_event.sql	View for all vaccination administration events.
325_t_adt_event.sql	All ADT events in the system.
325_v_adt_event.sql	View for all ADT events in the system.
390_1_tmp_events_drop.sql	Script to drop temporary tables.
501_t_pj_member.sql	All patient journey member records in the system.
501_v_pj_member.sql	View for all patient journey member records.
502_t_pj_procedure.sql	All patient journey procedure records in the system.
502_v_pj_procedure.sql	View for all patient journey procedure records.
503_t_pj_diagnosis.sql	All patient journey diagnosis records in the system.
503_v_pj_diagnosis.sql	View for all patient journey diagnosis records.
504_t_pj_laborder.sql	All patient journey lab order records in the system.
504_v_pj_laborder.sql	View for all patient journey lab orders records.
505_t_pj_labresult.sql	All patient journey lab result records in the system.

Script	Description
505_v_pj_labresult.sql	View for all patient journey lab result records.
506_t_pj_vitalsign.sql	All patient journey vital sign records in the system.
506_v_pj_vitalsign.sql	View for all patient journey vital sign records.
507_t_pj_socialhistory.sql	All patient journey social history records in the system.
507_v_pj_socialhistory.sql	View for all patient journey social history records.
508_t_pj_allergy.sql	All patient journey allergy records in the system.
508_v_pj_allergy.sql	View for all patient journey allergy records.
509_t_pj_encounter.sql	All patient journey encounter records in the system.
509_v_pj_encounter.sql	View for all patient journey encounter records.
510_t_pj_medicationhistory.sql	All patient journey medication history records in the system.
510_v_pj_medicationhistory.sql	View for all patient journey medication history records.
511_t_pj_immunizationhistory.sql	All patient journey immunization history records in the system.
511_v_pj_immunizationhistory.sql	View for all patient journey immunization history records.
541_pj_create_indexes.sql	Script to create indexes for all patient journey tables.
901_t_hv_availability.sql	Date of the Omni-Payer™ HealthViews build.

Script	**Description**
901_v_hv_availability.sql	View for the date of the Omni-Payer™ HealthViews build.
902_v_hv_release.sql	The version of Omni-Payer™ HealthViews that is installed.
941_create_indexes_script.sql	Script to create all indexes.
947_record_counts.sql	Generates record counts.
951_Grant_Select.sql	Generates user and group rights to the Omni-Payer™ HealthViews schema. For more information, see *Granting Users and Groups Access to Omni-Payer™ HealthViews* on page 23.

Granting Users and Groups Access to Omni-Payer™ HealthViews

The *951_Grant_Select.sql* script is used to generate user and group rights to the Omni-Payer™ HealthViews schema. It only needs to be run when new users or groups need access to Omni-Payer™ HealthViews, and not each time Omni-Payer™ HealthViews is built.

The first section of this script defines the users and groups that need to be given access to the schema. The output of this section is then used to create the actual grant statements, which are then run.

iWay

Reader Comments

In an ongoing effort to produce effective documentation, the Technical Content Management staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please share your suggestions for improving this publication and alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail:	Technical Content Management Information Builders, Inc. Two Penn Plaza New York, NY 10121-2898
Fax:	(212) 967-0460
Email:	*books_info@ibi.com*
Website:	*http://documentation.informationbuilders.com/connections.asp*

Name:____

Company:____

Address:____

Telephone:____________________Date:____________________

Email:____

Comments:

Reader Comments

Information Builders, Two Penn Plaza, New York, NY 10121-2898 **(212) 736-4433**