

# iWay

## **Omni-Patient™ Server Relational OnRamp User's Guide**

Version 2.3.x

Active Technologies, EDA, EDA/SQL, FIDEL, FOCUS, Information Builders, the Information Builders logo, iWay, iWay Software, Parlay, PC/FOCUS, RStat, Table Talk, Web390, WebFOCUS, WebFOCUS Active Technologies, and WebFOCUS Magnify are registered trademarks, and DataMigrator and Hyperstage are trademarks of Information Builders, Inc.

Adobe, the Adobe logo, Acrobat, Adobe Reader, Flash, Adobe Flash Builder, Flex, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2016, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

# Contents

<b>Preface.....</b>	<b>5</b>
Documentation Conventions.....	5
Related Publications.....	6
Customer Support.....	6
Help Us to Serve You Better.....	7
User Feedback.....	9
iWay Software Training and Professional Services.....	9
<b>1. Installing, Configuring, and Using the Relational On-Ramp.....</b>	<b>11</b>
Installation and Configuration.....	12
Database Connections.....	13
Representation of the Input Data.....	13
Tables.....	13
Table Mappings.....	14
Table Keys.....	14
Non-key Data Columns.....	15
Types of Loads Supported By the Ramp.....	17
Control Tables.....	18
Operational Details.....	20
Stand-Alone Loads.....	21
Differential Loads.....	21
Parallelism.....	27
Requirements of Batches.....	27
Source Codes.....	27
<b>Reader Comments.....</b>	<b>29</b>



# Preface

This documentation provides the details needed to understand and use the Omni-Patient™ Server Relational On-Ramp. It assumes an understanding of Omni concepts, OID input, and familiarity with relational databases. The abbreviation, ROR, will be used in this documentation to stand for Relational On-Ramp where convenient.

## How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
<b>1</b>	Installing, Configuring, and Using the Relational On-Ramp	Describes how to install, configure, and use the Omni-Patient™ Server Relational On-Ramp.

## 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.
<i>this typeface</i>	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.
<u>underscore</u>	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.

Convention	Description
{ }	Indicates two or three choices. Type one of them, not the braces.
	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](http://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.

<b>Platform</b>	
<b>Operating System</b>	
<b>OS Version</b>	
<b>JVM Vendor</b>	
<b>JVM Version</b>	

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

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

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

<b>iWay Adapter</b>	
<b>iWay Release Level</b>	

<b>iWay Patch</b>	
-------------------	--

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

<b>Request/Question</b>	<b>Error/Problem Details or Information</b>
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 <i>not</i> 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/consulting>.



# 1 Installing, Configuring, and Using the Relational On-Ramp

The Omni-Patient™ Server Relational On-Ramp is a set of relational tables and protocols used to feed data to an Omni System. It may be used as an alternative to, or in conjunction with, OID (XML instance document) input data.

This documentation provides the details needed to understand and use the Relational On-Ramp. It assumes an understanding of Omni concepts, OID input, and familiarity with relational databases. The abbreviation, ROR, will be used in this documentation to stand for Relational On-Ramp where convenient.

## Topics:

- ❑ Installation and Configuration
- ❑ Representation of the Input Data
- ❑ Types of Loads Supported By the Ramp
- ❑ Operational Details

## Installation and Configuration

**In this section:**

Database Connections

The Relational On-Ramp is a new component that installs automatically into Omni products and is manifested as a set of channels visible on the iWay Service Manager (ISM) Monitoring console.

Channel Name	Type	Role
ComparatorChannel	Internal	Compares tables
ComparatorDispatcherChannel	SQL	Looks for tables ready for compare
ComparatorPostProcessingAgentChannel	Internal	Reset tables for next differential
ComparatorPostProcessingChannel	SQL	Looks for tables done with conversion
RampDispatcherChannel	SQL	Looks for subjects ready for OID conversion
RampInputChannel	Ramp	Convert Ramp tables to OID
RampProcessingChannel	Internal	Route OIDs into Omni
RampTestLoader	FILE	Test load Ramp via UDL
FastPath_StoreAndMergeChannel	Internal	FastPath OID Input

**Note:** The channels are inactive at Omni start.

## Database Connections

The following parameters in DIB.properties determine the location of the On-Ramp tables. The default will be to use as the same connection as the model. Override this connection to access Ramp tables from another location.

```
ramp.ds.driver=
ramp.ds.url=
ramp.ds.username=
ramp.ds.password=
ramp.schema=
```

A complete set of On-Ramp tables will be created at the database and schema indicated above by the following command:

```
omni.sh db.create
```

### Note:

- ❑ Several sets of schemas are required by Differential and Fastpath loads.
- ❑ There are additional On-Ramp parameters that will be described at a later time.

## Representation of the Input Data

### In this section:

Tables  
Table Mappings  
Table Keys  
Non-key Data Columns

This section describes the various representations used of the input data.

### Tables

A subject of an Omni system is described by an IDS file. The IDS represents the subject as a hierarchy of documents: the top level subject document and child instance documents, or collections. For each such document, parent or child, there is a corresponding ROR table with name, nameOfSubjectOrCollection\_onr. For example, the Facility subject contains a Facility parent and a Facility\_Address child which are represented and loaded via the facility\_onr and facility\_address\_onr tables, respectively.

A row of an on Ramp table is analogous to a subject or a collection element in an OID.

- ❑ Like a subject or collection element in an OID, a row in an On-Ramp must correspond to *only one* instance of a record in a source system.

- ❑ While a child collection instance is related to its parent through its position in the XML structure of an OID, a child On-Ramp row is related to its parent through the relational database key.

There are additional tables delivered with the On-Ramp, `omnigen_ramp_control` and `omnigen_ramp_table_control`, whose purpose concerns the operation of the Ramp itself. These tables are updated by the integrator to signal that a particular batch (set of rows) is ready to be consumed by the Omni system.

The existing `omni.sh db.create` command will initialize empty On-Ramp tables.

## Table Mappings

Omni products provide an Excel workbook for each IDS subject, which describes the layout of the OID. The top level subject and each child collection are described by separate tabs. An analogous workbook describing the Relational On-Ramp columns is not currently available, so the integrator should use the existing OID workbook as a starting point. This, along with the information below concerning the representation of different types of OID data in the On-Ramp, should guide the mapping.

## Table Keys

Each on-Ramp table has a group of identifying columns, which taken together, map to the source record. These columns will be part of the primary key and are formed in the following way:

- ❑ Top level subjects
- ❑ Child collections
- ❑ Shared child collections
- ❑ Full primary key

For top level subjects, the identifying columns are:

- ❑ **source\_name.** Name of the source system (for example, `source_name = TestEpic`).
- ❑ **subject\_siid (Source Instance ID).** Unique Business identifier of the Subject within the named source system (for example, `facility_siid = 123`).

For child collections, the identifying columns are:

- ❑ **source\_name.** Same as the source name of the parent (for example, `source_name = TestEpic`).
- ❑ **top\_level\_subject\_siid.** Unique identifier of top level parent subject (for example, `facility_siid = 123`).

- ❑ **intermediate\_level\_collection\_siid [0-n].** Identifiers of each intermediate collection separating the child from the root subject. There is one siid per level. (for example, `facility_child_siid = CHILDTYPE1`, `facility_grandchild_siid = GRANDCHILDTYPE1`, and so on).
- ❑ **child\_siid.** Unique Business identifier of the child collection object within the hierarchy defined by the key (for example, a discriminator with respect to the parent or `facility_address_siid = PrimaryLocation`).

Equivalently, the identifying columns of a child collection may be defined recursively as the combination of the following:

- ❑ The identifying columns of the direct parent
- ❑ siid of the child

For shared child collections, the identifying columns are:

- ❑ **source\_name.** Same as the source name of the parent (for example, `source_name = TestEpic`).
- ❑ **parent\_subject.** The camel case IDS name of the direct parent collection (for example, `DiagnosisEvent`).
- ❑ **parent\_siid.** The unique identifier of the direct parent ('|' delimited if not top level).
- ❑ **child\_siid.** Unique Business identifier of the child collection object within the hierarchy defined by the key (for example, a discriminator with respect to the parent or `event_provider_siid = PrimaryProvider`).

Finally, the full primary key for any of these tables, whether top level subject, child collection or shared child collection, is comprised of the identifying columns described above and the `batch_id` column, which is present in each ROR table.

## Non-key Data Columns

The following are conventions for forming On-Ramp fields from their IDS names based in their type.

- ❑ **String, Double, Integer fields.** `<CamelCase>` becomes `<lower_case>`, as before. If these element names terminate with the reserved tokens, *Code*, *Link*, *Date*, *Time*, or *DateTime*, you can convert them to lower case appending a final underscore.
- ❑ **Date/Time fields (idsIidsDateTimeType).** It is recommended that date and time elements terminate with tokens *Date*, *Time*, or *DateTime* to indicate the desired precision. If these tokens are found, they are converted to lowercase in the On-Ramp field name using the usual transformation. If none of these tokens are found, then `_date_time` is appended to the IDS name.

- ❑ **Codes.** The convention is <CamelCase>+Code. Where followed, it is respected. Where not followed, the suffix `_code` is append. Then, three fields comprise the code reference.

```
<lower_case> + _code +_source  
<lower_case> + _code +_set  
<lower_case> + _code +_value
```

The element name changes from <CamelCase> to <lower\_case> and takes the following suffixes and links:

- ❑ **Sample 1.** Link is defined to refer to a single subject.

```
<lower_case> + _subjectName_ [if different than element name] + _link  
+ _source_name  
<lower_case> + _subjectName_ + [if different than element name] +_link  
+ _siid
```

- ❑ **Sample 2.** Link is defined to refer to one of several possible subjects. For each subject that the link may reference there will be a pair of columns.

```
<lower_case> + _subjectName + _link +_source_name  
<lower_case> + _subjectName_ + _link +_siid
```

**Note:** The integrator should populate up to one pair of `source_name/siid` (or exactly one if the link is not optional) and the On-Ramp must reject as an error records with link columns populated for more than one subject.

- ❑ **Group Names.** Elements within groups work unchanged.

```
<CamelCase>  
becomes  
<grp_id> + <lower_case>
```

### Additional Columns

- ❑ **batch\_id.** This is part of the key for each On-Ramp table. Every On-Ramp row is associated with a input batch.
- ❑ **onr\_created\_datetime.** Not currently implemented. If the integrator sequences entries in a batch by this field, the On-Ramp may in the future use it for recovery.



## Types of Loads Supported By the Ramp

### In this section:

#### Control Tables

The Relational On-Ramp supports three different load paradigms:

- ❑ **Stand-Alone.** A stand-alone load processes a single batch of Ramp tables which needs to be populated in full before the load is submitted.
  - ❑ Integrator supplies all on Ramp tables for a subject prior to load of batch.
  - ❑ May be full initial load or incremental load (delta provided by the integrator).
  - ❑ No support for connecting orphaned persons to parents.
  - ❑ No life-cycle management of loads.
- ❑ **Differential.** A Differential load processes two consecutive batches of Ramp tables, denoted as lastbatch and newbatch, which are compared to form a difference set and submitted to Omni. Ramp tables for the newbatch may be presented for processing individually as they become available.
  - ❑ Integrator updates ACTIVE column in omnigen\_ramp\_table\_control to indicate which tables are part of integration set (configuration).
  - ❑ Integrator updates batch\_type to 'DIFFERENTIAL' for each active column.
  - ❑ Integrator supplies individual tables as ready.
  - ❑ Differential between current load and prior load is calculated by the Ramp.
  - ❑ System orchestrates when steps proceed depending on when the individual Ramp tables are ready.
  - ❑ Life-cycle support for Ramp tables. After completion, an archive is updated and the Ramp is returned to ready state for subsequent processing.
  - ❑ Orphaned Person support
- ❑ **FastPath.** A FastPath load is a high performing initial load of data for a subject to an Omni system. The Omni database is populated directly bypassing regular OID processing.
  - ❑ Integrator updates ACTIVE column in omnigen\_ramp\_table\_control to indicate which tables are part of integration set (configuration).
  - ❑ Integrator updates batch\_type to 'FASTPATH' for each active column.

- ❑ Initial load only.
- ❑ Data must require no cleansing. Source codes must be preloaded.

## Control Tables

Control tables are used to initiate loads and monitor the status and results.

### **omnigen\_ramp\_control**

This table is used by the integrator to initiate Stand-Alone loads via the On-Ramp. A new row is added to signal that a batch for a subject is ready for processing. The same row will then collect statistics about the processing that occurred.

The omnigen\_ramp\_control has the following columns:

Field	Description
batch_id	Supplied by Integrator. An identifier which binds a set of tables into a single logical load.
batch_type	Populated by Omni. Internal to use of FASTPATH processing.
batch_options	Populated by Omni. Derived from onramp_table_control for subject.
subject	Supplied by Integrator. The top level subject for which this batch is to run. Subject is in the CamelCase format that appears in the name attribute of the associated IDS.
created_datetime	Supplied by Integrator. Should be a timestamp for the batch submission. The On Ramp Reader will process batches in created time order.
total_records	Populated by Omni. The total number of rows across tables for subject in a batch. This may be a useful confirmation that the intended tables are being processed.
total_output	Populated by Omni. Count of intermediate OIDs generated by the batch.
processed_datetime	Populated by Omni. Timestamp indicating when batch processing is complete. NULL if processing in progress.
error	Populated by Omni. If the On-Ramp could not continue, then an error message is supplied

omnigen\_ramp\_control key: batch\_id, subject

### **omnigen\_ramp\_table\_control**

This table is used by the integrator to initiate Differential loads using the On-Ramp. A new row is updated to signal that an individual Ramp table is ready for processing. The same row will then collect statistics about the processing that occurred for that table.

<b>Field</b>	<b>Description</b>
subject	Supplied by Integrator. The top level subject to which table_name belongs. Separate row for each subject in the case of shared children. The subject is in the CamelCase format that appears in the name attribute of the associated IDS.
table_name	Supplied by Integrator. Name of the table to appear in a Differential load. The name is lower case with sections separated by underscores.
batch_id	Supplied by Integrator. An identifier which binds a set of tables into a single logical load.
batch_type	Supplied by Integrator. You can set this to <i>FASTPATH</i> or <i>DIFFERENTIAL</i> to indicate the type of batch.
batch_options	Supplied by Integrator. You can set this to <i>SKIPCONVERT</i> to indicate a FASTPATH copy is to occur without Subject OID conversion.
active	Supplied by Integrator. Set to Y if table is participating in Differential load. Otherwise, you can set this to N.
state	Supplied by Integrator/Updated by Omni. You can set by Integrator to GO for processing as path of a batch. It is used by Omni to manage internal states completing with READY or ERROR.
prior_state	Populated by Omni. Internal. For diagnostic use.
error_info	Populated by Omni. Text information associated with error.
integration_start_datetime	Populated by Integrator. When table is set to GO
integration_end_datetime	Populated by Omni. Not currently used.

Field	Description
compare_start_datetime	Populated by Omni. Start of table comparison.
compare_end_datetime	Populated by Omni. End of comparison.
convert_start_datetime	Populated by Omni. NOT used.
convert_end_datetime	Populated by Omni. End of OID conversion.
copy_start_datetime	Populated by Omni. Start of FastPath copy.
copy_end_datetime	Populated by Omni. End of FastPath copy.
postprocess_start_datetime	Populated by Omni. Start of Postprocessing.
postprocess_end_datetime	Populated by Omni. End of PostProcessing / Error.
record_count	Populated by Omni. Not currently used.
incremental_record_count	Populated by Omni. Number of new or changed records in table.
oid_count	Populated by Omni. Not currently used.

omnigen\_ramp\_table\_control key: subject, table\_name

For all the load paradigms, the Ramp control tables are located in the default schema of the ramp.datasources file.

## Operational Details

**In this section:**

- Stand-Alone Loads
- Differential Loads
- Parallelism
- Requirements of Batches
- Source Codes

This section describes the operational details.

## Stand-Alone Loads

A batch of Ramp tables are converted to OIDs. The stand-alone load may be a full initial system load or a subsequent incremental load. The top level subject is the unit of submission.

The basic usage sequence is:

- ❑ Integrator populates ramps tables (\*\_onr), grouping records to be processed together with the same batch\_id.
- ❑ Integrator executes a request to read the On-Ramp tables by writing a row or rows to the omnigen\_ramp\_control table, populating the following columns:
  - ❑ **Batch\_id.** Identifier of the batch. Used as a WHERE condition on \_onr tables.
  - ❑ **Subject.** Top level subject.
  - ❑ **Created\_date.** Start of load (for reference only).
  - ❑ **Processed\_datetime.** Set to NULL to initiate - filled in by converter after.
- ❑ The Relational On-Ramp reads and processes the batch. The On-Ramp Converts the tables into a sequence of OID documents and feeds them to an Omni system using FileSystemIDSReader, OmniInterface table, or Internal Queue.
- ❑ The number of records read and processed, and any available error information is written to omnigen\_ramp\_control.

**Note:** While a batch is being processed, no record for that batch\_id should be altered.

The following example illustrates submitting a batch for Facility with the output on completion:

### omnigen\_ramp\_control

Batch_id	Subject	Created_datetime	Total_records	Total_output	Processed_datetime	error
1	Facility	"2015-11-03 13:39:38.351385"				
1	Facility	"2015-11-03 13:39:38.351385"	6	6	"2015-11-03 13:39:41.014"	

## Differential Loads

Two sets of Ramp tables representing consecutive full loads are compared. A difference set is computed and these are converted to OIDs. The on-Ramp table is the unit of submission.

Five sets of on-Ramp tables are used to support the Differential load life cycle. These must be created in five different schemas within the Ramp data source.

Set	Use	Schema Name/Derivation
ramp	Input to OID converter/Omni; Ramp control tables.	set by ramp.schema in DIB.properties
incrementalbatch	Output of Differential computation.	set by comparator.batch.schema.name in DIB.properties
newbatch	New batch input to compare.	newbatch (hardcoded)
lastbatch	Last batch input compare.	lastbatch (hardcoded)
archive	Collected batches prior to lastbatch.	archive (hardcoded)

The following list describes the sequence of the Differential load life-cycle:

- ❑ The prior complete load is already in the lastbatch schema tables (or lastbatch is empty if this is initial load).
- ❑ Newbatch Ramp tables are populated with new full load.
- ❑ Incremental batch is calculated from lastbatch and newbatch.
- ❑ If incrementalbatch schema is the same as ramp schema, then conversion to OID proceeds immediately. Otherwise, manual interventions are required.
- ❑ Lastbatch is appended to archive, newbatch is copied to lastbatch. Newbatch and incrementalbatch are cleared.

The basic usage sequence is:

- ❑ Prepare omnigen\_ramp\_table\_control to identify tables included in the Differential load.
- ❑ Set state of row to GO and batch\_type='DIFFERENTIAL' when each newbatch table is ready.
- ❑ Differential load is complete when all ACTIVE='Y' tables have been set to GO by integrator and then returned to READY by the system.

Rules for setting rows active are found in omnigen\_ramp\_table\_control.

It is important that the ACTIVE is set to Y with the correct table\_name subject pairs.

- ❑ Each table being used as part of the integration should be represented by a row with its enclosing subject with ACTIVE='Y'.
- ❑ A table not being used may either be omitted from omnigen\_ramp\_table\_control or present with ACTIVE='N'.
- ❑ A table that is a shared child collection (IDS type="instance" access="shared", for example, EventProvider) should have a row, ACTIVE='Y' for each parent subject that contains it.
- ❑ For Person or other subjects that are referenced as type="document":
  - ❑ All tables within Person that are part of the batch need a row, active='Y', with subject="Person"
  - ❑ For each subject within the batch that references Person, you will need to link the Person parent table, person\_onr, with the referencing subject (for example, table\_name='person\_onr', subject='Patient', active='Y')

The following is an example for a batch that contains the following:

- ❑ facility, facility\_address, but not facility\_location.
- ❑ event\_providers for DiagnosisEvents, AdmissionEvents, and DischargeEvents but not TransferEvent.
- ❑ Persons and Person names for Patient and Provider and patient preferredProvider.

Subject	Table_name	Batch_id	Active	State
AdmissionEvent	event_provider_onr	2	Y	READY
DiagnosisEvent	event_provider_onr	2	Y	READY
DischargeEvent	event_provider_onr	2	Y	READY
Facility	facility_onr	2	Y	READY
Facility	facility_address_onr	2	Y	READY
Facility	facility_location_onr	2	N	READY
Patient	patient_onr	2	Y	READY
Patient	patient_preferred_provider_onr	2	Y	READY

Subject	Table_name	Batch_id	Active	State
Patient	person_onr	2	Y	READY
Person	person_name_onr	2	Y	READY
Person	person_onr	2	Y	READY
Provider	provider_onr	2	Y	READY
Provider	person_onr	2	Y	READY
TransferEvent	event_provider_onr	2	N	READY

### Rules for submitting a batch

- ❑ One batch must be complete before the next one is started.
- ❑ As tables become available, set them to GO.
- ❑ For EventProvider and EventNote, set all active subjects to GO.
- ❑ For type="document" parent node (for example, person\_onr) Set person\_onr/Person to GO. Do not set Person\_onr/Patient to GO because this is managed by the system.

For example, Newbatch event\_provider\_onr table (for all parent data) facility\_address\_onr, person\_onr, and person\_name\_onr are populated and should be processed.

Subject	Table_name	Batch_id	Active	State
AdmissionEvent	event_provider_onr	2	Y	GO
DiagnosisEvent	event_provider_onr	2	Y	GO
DischargeEvent	event_provider_onr	2	Y	GO
Facility	facility_onr	2	Y	READY
Facility	facility_address_onr	2	Y	GO
Facility	facility_location_onr	2	N	READY
Patient	patient_onr	2	Y	READY
Patient	patient_preferred_provider_onr	2	Y	READY



Subject	Table_name	Batch_id	Active	State
Patient	person_onr	2	Y	GO
Person	person_name_onr	2	Y	GO
Person	person_onr	2	Y	READY
Provider	provider_onr	2	Y	READY
Provider	person_onr	2	Y	READY
TransferEvent	event_provider_onr	2	N	READY

### FastPath Loads

A FastPath load is a speed-optimized processing pathway for initial data loads for subjects that are non-mastered and require no cleansing. The on-Ramp table is the unit of submission.

The FastPath load occurs in the following two stages.

- ❑ For each Ramp table, SQL which INSERTs the contents into the corresponding model table(s), is constructed and executed. Omnigen\_ramp\_table\_control gives the table and subject names. If a Ramp table name is active for a single subject, then the model table is truncated prior to the INSERT. If the table is shared among subjects, then DELETE and INSERT occur separately for records belonging to each subject.
- ❑ OIDs are generated for each subject after all the subject tables have been copied. These OIDs are fed to a special channel that performs a reduced subset of Omni processing. Currently, this is only to populate omni\_working table.

**Note:** Certain tables (for example, Event Types) map to two model tables.

FastPath load uses the following data sources and schemas, which are specified in DIB.properties.

Data Source	Schema	Use
ramp.fastpath.datasource	ramp.fastpath.source.schema	Source data for FastPath copy (stage 1). Also source data for OID conversion (stage 2) if ramp.fastpath.convertinplace = true.

Data Source	Schema	Use
ramp.fastpath.datasources	ramp.fastpath.target.schema	Target data (instance model) for FastPath.
ramp.datasources	default	Location of Ramp control tables. Source data for OID conversion (stage 2) if ramp.fastpath.convertinplace = false.
omni.datasources	default	Location of OmniWorking Table.

The following steps describe the basic usage sequence.

1. Prepare `omnigen_ramp_table_control` to identify On-Ramp tables included in a FastPath load.
2. Set state of row to `GO` and `batch_type='FASTPATH'` when each table is ready.

Fastpath load is complete when all `ACTIVE='Y'` tables have been set to `GO` by the integrator and then returned to `READY` by the system.

The rules for setting rows active in `omnigen_ramp_table_control` are the same as those for a Differential load.

### Options for FastPath

- ❑ **SKIPCONVERT.** Set `batch_options` column in `omnigen_ramp_table_control` to `SKIPCONVERT` to suppress OID conversion. A subject will not be converted if any of tables included within the subject indicate `SKIPCONVERT`.
- ❑ **ramp.fastpath.datasources.type.** Set the database type of the FastPath data source in `DIB.properties` to generate the correct platform specific. Valid choices are `POSTGRES`, `DB2`, and `MSSQL`.
- ❑ **ramp.fastpath.convertinplace.** Set in `DIB.properties` to indicate whether the same data source used for copy may be used for conversion. If not true, the integrator must perform the following steps:
  1. Set state of row to `GO`, `batch_type='FASTPATH'` for each FastPath On-Ramp table. When the copy is complete, the system alters the state to `PENDINGTRANSFER`.
  2. Integrator copies or recreates the On-Ramp tables in the default schema of `rampdatasources`.
  3. Integrator changes the `STATE` from `PENDINGTRANSFER` to `COPIED`.

The system will proceed to convert all copied subjects and return the state of the Ramp tables to *READY* when complete.

The following list shows the FastPath limitations:

- ❑ No support for promotions (for example, address in facility).
- ❑ Requires Source Codes to be preloaded.

### Parallelism

The Relational On-Ramp will process batches for multiple top level subject concurrently. Multiple batches for the same top level subject will be processed serially.

### Requirements of Batches

The batch is the unit of processing for the Relational On-Ramp. A batch is launched by writing a row in `omnigen_ramp_control` indicating the `batch_id` and `subject`. Each record in the on-Ramp table must correspond to a unique record in the source system. This is enforced by the On-Ramp table keys. It is likely, that performance will be improved if the Integrator provided full hierarchies of subjects, but the Ramp does not require this. If a child collection is encountered whose parent (identified by the key) is not in the batch. The appropriate *stub* parent will be created.

### Source Codes

Source Codes and Source Code Sets are loaded via the On-Ramp tables like other subjects. The top level subject to use when submitting a batch with either or both is `SourceCodeSet`. However, the IDS structure for `SourceCode` and `SourceCodeSet` is slightly different from standard subjects, so the following considerations apply.

- ❑ Duplicate naming of fields
  - ❑ **source\_code\_set\_onr.** `Source_code_set_siid` and `code_set_name` must be populated duplicate values of the code set name.
  - ❑ **source\_code\_onr.** `Source_code_siid` and `source_code` must be populated duplicate values of the source code.
- ❑ Source Codes reference their sets as part of the key. The records for these sets (from `source_code_set_onr`) must either be included in the On-Ramp batch with the source codes or have already been loaded. *Stub* CodeSets will be created if Orphan processing is enabled.

- ❑ Parent Codes need to be loaded before codes that reference them as parents. This is the same ordering requirement that OIDs have, but expressed in the order of On-Ramp batches.

**Orphan Support**

**Note:** Deletions are not currently supported in either Stand-Alone or Differential loads.

## 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](mailto:books_info@ibi.com)

**Website:** <http://documentation.informationbuilders.com/connections.asp>

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

Telephone: \_\_\_\_\_ Date: \_\_\_\_\_

Email: \_\_\_\_\_

Comments:

# **Reader Comments**