

**Oracle® Retail Omnichannel Cloud Data Service**

Implementation Guide

Release 16.0.030

**F12556-02**

August 2019

F12556-02

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Owen Horne

Contributing Author:

Contributor:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

#### **Value-Added Reseller (VAR) Language**

##### **Oracle Retail VAR Applications**

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

- (i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.
- (iii) the software component known as **Access Via**<sup>™</sup> licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.
- (iv) the software component known as **Adobe Flex**<sup>™</sup> licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications.

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

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



---

---

# Contents

<b>Send Us Your Comments</b> .....	ix
<b>Preface</b> .....	xi
Audience .....	xi
Documentation Accessibility .....	xi
Related Documents .....	xi
Customer Support .....	xi
Improved Process for Oracle Retail Documentation Corrections .....	xii
Oracle Retail Documentation on the Oracle Technology Network .....	xiii
Conventions .....	xiii
<b>1 Introduction</b>	
<b>2 Technical Design</b>	
OCDS Topology .....	2-1
Scaling OCDS .....	2-1
WebLogic Server Cluster Concepts .....	2-2
BDI Job Admin on Cluster .....	2-2
RIB Injector on Cluster .....	2-2
ORDS on Cluster .....	2-3
The OCDS Database .....	2-3
Tablespaces .....	2-3
Schemas .....	2-3
Transactional Schema .....	2-3
Timestamp Control Columns.....	2-4
Logging .....	2-4
Interface Schema .....	2-4
<b>3 BDI Job Admin</b>	
BDI Interfaces .....	3-1
Scheduling BDI Data .....	3-2
Scheduled .....	3-2
One-Time Only .....	3-2
WebLogic Users and Groups.....	3-3
Groups .....	3-3

Users.....	3-3
<b>Changing User Passwords</b> .....	3-3
<b>Logging</b> .....	3-4
<b>URL patterns</b> .....	3-4
<b>Job Admin User Interface</b> .....	3-4
<b>4 RIB Injector</b>	
<b>RIB Families</b> .....	4-1
<b>Logging</b> .....	4-3
<b>WebLogic Users and Groups</b> .....	4-3
Groups .....	4-4
Users.....	4-4
<b>Changing User Passwords</b> .....	4-4
<b>5 ORDS Web Services</b>	
<b>Performance Tuning</b> .....	5-1
<b>REST Services</b> .....	5-1
Authentication .....	5-1
Endpoints .....	5-1
<b>WebLogic Users and Groups</b> .....	5-2
Groups .....	5-2
Users.....	5-2
<b>Changing User Passwords</b> .....	5-2
<b>6 Maintenance</b>	
<b>Data Retention</b> .....	6-1
Purging Deleted Data .....	6-1
<b>A Appendix A: REST Web Service API</b>	
<b>diffs: Get Item Differentiators</b> .....	A-1
Parameters.....	A-1
Response (items).....	A-2
Example Data.....	A-3
<b>diff/group: Get Item Differentiator Group</b> .....	A-3
Parameters.....	A-4
Response (items).....	A-5
Example Data.....	A-5
<b>item: Get Item</b> .....	A-6
Parameters.....	A-6
Response (items).....	A-7
Example Data.....	A-14
<b>item/dimensiontype: Get Item Dimension Type</b> .....	A-15
Parameters.....	A-15
Response (items).....	A-16
Example Data.....	A-17
<b>item/dimensionvalue: Get Item Dimension Value</b> .....	A-17

Parameters.....	A-18
Response (items).....	A-19
Example Data.....	A-19
<b>item/image: Get Item Image</b> .....	A-19
Parameters.....	A-20
Response (items).....	A-21
Example Data.....	A-22
<b>item/itemlocation: Get Item Location</b> .....	A-22
Parameters.....	A-22
Response (items).....	A-24
Example Data.....	A-26
<b>item/price: Get Item Price</b> .....	A-27
Parameters.....	A-27
Response (items).....	A-28
Example Data.....	A-29
<b>item/relateditem: Get Related Item</b> .....	A-29
Parameters.....	A-30
Response (items).....	A-31
Example Data.....	A-33
<b>location/retailstore: Get Retail Store</b> .....	A-34
Parameters.....	A-34
Response (items).....	A-35
Example Data.....	A-36
<b>location/warehouse: Get Warehouse</b> .....	A-37
Parameters.....	A-37
Response (items).....	A-38
Example Data.....	A-40
<b>merchhier: Get Merchandise Hierarchy</b> .....	A-40
Parameters.....	A-41
Response (items).....	A-42
Example Data.....	A-43
<b>metadata-catalog: Get API Catalog</b> .....	A-43
Parameters.....	A-44
Response (items).....	A-44
Example Data.....	A-44
<b>orghier: Get Organization Hierarchy</b> .....	A-45
Parameters.....	A-45
Response (items).....	A-47
Example Data.....	A-48
<b>orghier/descendant: Get Organization Hierarchy Node Descendant</b> .....	A-48
Parameters.....	A-48
Response (items).....	A-49
Example Data.....	A-50
<b>refreshdate: Get Data Refresh Date</b> .....	A-51
Parameters.....	A-51
Response (items).....	A-52
Example Data.....	A-52

<b>inventory/future: Get Future Inventory</b> .....	A-52
Parameters.....	A-53
Response (items).....	A-54
Example Data.....	A-54
<b>inventory/store: Get Store Inventory</b> .....	A-55
Parameters.....	A-55
Response (items).....	A-56
Example Data.....	A-56
<b>inventory/warehouse: Get Warehouse Inventory</b> .....	A-57
Parameters.....	A-57
Response (items).....	A-58
Example Data.....	A-59



---

---

# Send Us Your Comments

Oracle® Retail Omnichannel Cloud Data Service Implementation Guide, 16.0.030

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

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

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

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

---

---

**Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

---

---

Send your comments to us using the electronic mail address: [retail-doc\\_us@oracle.com](mailto:retail-doc_us@oracle.com)

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

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

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



---

---

# Preface

The *Oracle® Retail Omnichannel Cloud Data Service Implementation Guide* provides information about the processing of the Oracle Omnichannel Cloud Data Service (OCDS) data hub.

## Audience

This guide is for technical personnel who configure, maintain and support, or use Oracle Retail Xstore Office.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

For more information, see the Oracle Retail documentation set.

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

## Customer Support

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

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received

- Screen shots of each step you take

## Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

# Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following web site:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

## Conventions

The following text conventions are used in this document:

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



---

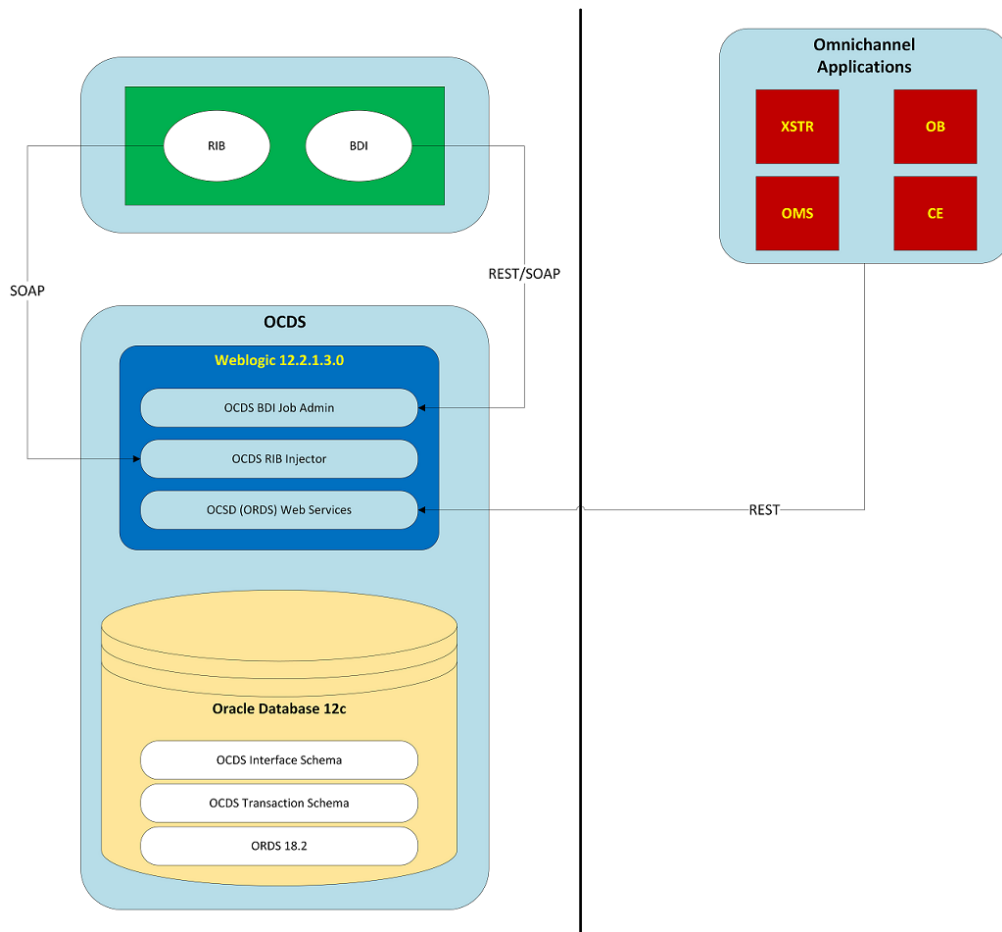
---

## Introduction

Oracle Omnichannel Cloud Data Service (OCDS) is a data hub, enabling Oracle Retail Merchandising and Pricing applications to share foundation data with Oracle Retail Omnichannel applications. OCDS is composed of three major components:

- BDI Batch Job Admin - Enables the flow of data into OCDS using Oracle Bulk Data Integration (BDI) technology. Job Admin has a User Interface (UI) to support the management of BDI batch Jobs.
- RIB Injector - Enables the flow of data into OCDS from the Oracle Retail Integration Bus (RIB).
- ORDS - Enables the data contained in OCDS to be accessed by Omnichannel Applications through the use of RESTful web services.

**Figure 1–1 OCDS Components**



---

---

**Note:** The flow of Promotion data into and out of OCDS is not currently supported. The Promotion Offer BDI interface should not be scheduled or manually invoked.

---

---

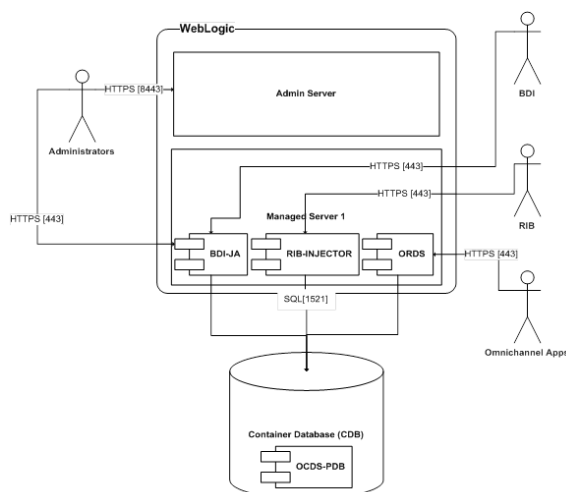


This chapter deals with the technical design of OCDS.

## OCDS Topology

The diagram below illustrates the basic deployment topology for OCDS. Alternatively, each OCDS component can be hosted in its own WebLogic Managed Server.

**Figure 2–1 Basic Deployment**



- **BDI-JA:** OCDS (BDI) Job Admin is the interface between the Oracle Retail Bulk Data Integration and OCDS, enabling BDI data to flow into the OCDS database.
- **RIB-INJECTOR:** OCDS (RIB) Injector is the interface between RIB infrastructure and OCDS; it listens for SOAP-based RIB messages containing incremental changes to data initially populated through BDI.
- **ORDS:** The OCDS (ORDS) Web Service exposes the data managed by OCDS to Omnichannel applications.

## Scaling OCDS

Oracle provides many clustering solutions and options; those relevant to OCDS are Oracle database clusters and WebLogic Server clusters. Clustering directly addresses availability, scalability, recoverability requirements which are very attractive to a business. In reality though it is a tradeoff, a clustered system increases complexity, is

normally more difficult to manage and secure, so one should evaluate the pros and cons before deciding to use clustering.

## WebLogic Server Cluster Concepts

A WebLogic Server cluster consists of multiple WebLogic Server managed server instances running simultaneously and working together to provide increased scalability and reliability. A cluster appears to clients to be a single WebLogic Server instance. The server instances that constitute a cluster can run on the same machine, or be located on different machines. You can increase a cluster's capacity by adding additional server instances to the cluster on an existing machine, or you can add machines to the cluster to host the incremental server instances. Each server instance in a cluster must run the same version of WebLogic Server.

In an active-passive configuration, the passive components are only used when the active component fails. Active-passive solutions deploy an active instance that handles requests and a passive instance that is on standby. In addition, a heartbeat mechanism is usually set up between these two instances together with a hardware cluster (such as Sun Cluster, Veritas, RedHat Cluster Manager, and Oracle CRS) agent so that when the active instance fails, the agent shuts down the active instance completely, brings up the passive instance, and resumes application services. In an active-active model all equivalent members are active and none are on standby. All instances handle requests concurrently. An active-active system generally provides higher transparency to consumers and has a greater scalability than an active-passive system. On the other hand, the operational and licensing costs of an active-passive model are lower than that of an active-active deployment.

---

---

**Note:** See the Oracle® Fusion Middleware Using Clusters for Oracle WebLogic Server documentation for more information.

[https://docs.oracle.com/cd/E15523\\_01/web.1111/e13709/toc.htm](https://docs.oracle.com/cd/E15523_01/web.1111/e13709/toc.htm)

---

---

## BDI Job Admin on Cluster

The OCDS BDI Job Admin is, in Oracle® Retail Bulk Data Integration terminology, a BDI edge application. BDI edge apps can be deployed into in an active-active cluster environment to achieve better throughput.

For more information see the *Oracle® Retail Bulk Data Integration Implementation Guide*.

## RIB Injector on Cluster

The RIB Injector is a Web Service consumer of SOAP messages.

The RIB architecture is designed for active-passive clusters only, so there can only be one injector end points on the consuming side. However, through-put can be optimized by customizing the RIB's multi-channel feature that allows parallel threads to send messages to a consumer, such as OCDS. The producer-side can be setup with multiple channels; for example, there would be 10 subscriber adapters in play sending messages in parallel to the OCDS Injector end point. The # of channels can be configured and will typically match how fast the consuming application can consume the messages.

See chapter 10 of the *Oracle® Retail Integration Bus Operations Guide* for more information.

## ORDS on Cluster

The ords.war file can be deployed onto more than one managed server. However, please note that the deploy procedure does not copy or recreate the ORDS dependent files and directories (stored in config/ords) onto each managed server. These directories and files must be made available on each managed server.

For more detailed information see *Oracle REST Data Services Installation, Configuration, and Development Guide*.

## The OCDS Database

The OCDS database contains two distinct schemas, one serves as the interface to the Bulk Data Integration, the other contains the data served to OCDS web service clients.

## Tablespaces

Tablespaces are a collection of logical storage units in the database. The tablespace to schema association is determined by database administrators at install time.

## Schemas

Schema objects are the logical structures that directly refer to the database's data. Schema objects include structures like tables, procedures, and indexes.

**Table 2–1 Schema Objects**

Type	Database User	Description
BDI Interface	ocds_ifc	The database contains BDI Interface tables into which the Oracle Bulk Data Integration inserts Merchandising and Pricing data. The content of BDI Interface tables is only changed by BDI activity.
OCDS Transactional	ocds_txn	The database contains transactional tables that function as a living snapshot of the latest Merchandising and Pricing data. The contents of Transactional tables are initially populated by BDI. In addition, Transactional tables are modified when incremental changes to Merchandising data are received from RIB messages.

### Transactional Schema

The *transactional* schema contains tables necessary for serving the Enterprise Data needs of Omnichannel applications. Most transactional tables are initially populated by BDI activity. In addition, many tables are modified by the contents of RIB messages.

Despite what this schema's name might suggest, the Transactional Schema does not contain any data retailers commonly refer to as "transaction data". In OCDS terms, the word "transactional" distinguishes how the contents of this schema's tables change frequently over time, primarily as RIB messages are consumed, whereas the Interface

Schema's table contents are a static snapshot of exported data from Merchandising and Pricing systems.

The core Transactional schema tables are initially populated as part of the inbound BDI flow managed by Job Admin. After BDI Interface tables are loaded successfully from BDI, a BDI "importer" copies data from Interface schema tables to Transactional schema tables.

The core Transactional schema tables contain, in many cases, only a subset of the columns of their corresponding table in the Interface schema. Only columns useful to Omnichannel applications are included in the Transactional schema's tables.

**Timestamp Control Columns** Transactional schema tables that contain data also communicated from RIB messages include the use of timestamp control columns. These columns are used by the OCDS web service API in calculating an "action" value. Action values provide a hint to the calling application how the data should be handled. Actions have values such as INSERT, UPSERT, UPDATE, and DELETE.

The timestamp control columns are:

**Table 2-2 Timestamp Control Columns**

Column Name	Description
LAST_CREATE	Last created
INIT_CREATE	Initially created
LAST_UPDATE	Last updated
LAST_DELETE	Last deleted (NULL if not deleted)

**Logging** The Transactional schema contains two diagnostic logging-related tables: LOG\_ENTRY and LOG\_SEVERITY. The LOG\_SEVERITY table can be used to turn on and off the ERROR, INFO and DEBUG log messages that can be written to the LOG\_ENTRY table by OCDS database procedures in the transactional schema.

### Interface Schema

The *interface* schema contains all of the inbound-interface tables necessary for receiving Merchandising and Pricing data from the Bulk Data Integration (BDI) infrastructure. OCDS is a receiving-only BDI Edge Application. Up-stream BDI components have responsibility for sending data to OCDS, and into the interface schema.

---



---

## BDI Job Admin

BDI Job Admin is a web application and services responsible for management of the receiving of data into OCDS from BDI. In as much as BDI Job Admin is a component of OCDS, it is also a component of the enterprise infrastructure that enables OCDS to receive Merchandising and Pricing data.

### BDI Interfaces

OCDS (BDI) Job Admin is the interface between BDI infrastructure and OCDS, enabling BDI data to flow into the OCDS database. The following table lists the BDI Interfaces supported by OCDS for which data is exposed through REST web service resources.

**Table 3–1 BDI Interfaces supported by OCDS**

<b>Interface Module</b>	<b>BDI Type</b>
Clearance_Tx	Transactional
COFutureAvail_Tx	Foundational
DiffGrp_Fnd	Foundational
Diff_Fnd	Foundational
InvAvailStore_Tx	Foundational
InvAvailWh_Tx	Foundational
ItemHdr_Fnd	Foundational
ItemImage_Fnd	Foundational
ItemLoc_Fnd	Foundational
MerchHier_Fnd	Foundational
OrgHier_Fnd	Foundational
PriceChange_Tx	Transactional
RelatedItem_Fnd	Foundational
StoreAddr_Fnd	Foundational
Store_Fnd	Foundational
VatItem_Fnd	Foundational
Vat_Fnd	Foundational
WhAddr_Fnd	Foundational
Wh_Fnd	Foundational

## Scheduling BDI Data

OCDS receives data from two sources, BDI and RIB. For data families where RIB messages trickle incremental changes to OCDS, a one-time BDI job provides OCDS with an initial load. For data families without RIB support, scheduled BDI jobs provide OCDS with recurring refreshes.

**Table 3–2 Recommended BDI Refreshes for OCDS**

<b>BDI Family</b>	<b>BDI Refresh</b>
Clearance Price	SCHEDULED
Diff Group	ONE-TIME ONLY
Diff	ONE-TIME ONLY
Inv Avail Store	SCHEDULED
Inv Avail Warehouse	SCHEDULED
Inv Future Available	SCHEDULED
Item Header	ONE-TIME ONLY
Item Image	ONE-TIME ONLY
Item Location	ONE-TIME ONLY
Item Supplier	ONE-TIME ONLY
Merchandise Hierarchy	ONE-TIME ONLY
Organization Hierarchy	ONE-TIME ONLY
Price Change	SCHEDULED
Related Item	ONE-TIME ONLY
Store	ONE-TIME ONLY
VAT	SCHEDULED
VAT Item	ONE-TIME ONLY
Warehouse	ONE-TIME ONLY

### Scheduled

BDI families not utilizing RIB for incremental changes should be scheduled using the BDI Scheduler at intervals appropriate for the BDI family.

### One-Time Only

BDI families where initial load of data is provided by BDI, and incremental changes are provided by RIB, should not be scheduled using the BDI Scheduler. The only time these OCDS should receive a subsequent BDI, after the initial load, is if the OCDS data is determined to be unreliable.

**Important:** Note that refreshing OCDS one-time only families may force downstream system, such as Xstore, to purge and completely refill all databases. Understanding how Omnichannel applications behave when OCDS is refreshed is extremely important when evaluating the decision to refresh "One-Time Only" data in OCDS from BDI.

## WebLogic Users and Groups

WebLogic security realms provide a mechanism for protecting WebLogic resources. Users are entities that can be authenticated in a security realm. Users are organized into Groups that can have different levels of access to WebLogic resources.

This section lists the default security realm's Groups and Users used by the BDI Job Admin component.

See *Monitoring Batch Jobs Using BDI Job Admin in Oracle® Retail Bulk Data Integration Implementation Guide* for information about Job Admin User roles.

### Groups

**Table 3–3 WebLogic Groups**

Name	Description	Component
BdiEdgeOcdsJobAdminGroup	User who can create, run and monitor Jobs	BDI Job Admin
BdiEdgeOcdsJobMonitorGroup	Users who can run and monitor jobs	BDI Job Admin
BdiEdgeOcdsJobOperatorGroup	Users who can monitor jobs	BDI Job Admin

### Users

**Table 3–4 WebLogic Users**

Type	Group	Description	Component
Job Administrators	BdiEdgeOcdsJobAdminGroup	Job Administrators	BDI Job Admin
Job Operators	BdiEdgeOcdsJobOperatorGroup	Job Operators	BDI Job Admin
Job Monitors	BdiEdgeOcdsJobMonitorGroup	Job Monitors	BDI Job Admin

## Changing User Passwords

Job Admin user passwords are authenticated by the default Authentication Provider configured for the security realm of the WebLogic domain in which Job Admin is hosted. The WebLogic security architecture supports a variety of types of Authentication Providers, including custom. The out-of-the-box Authentication Provider is the WebLogic Authentication provider, also known as the DefaultAuthenticator, which accesses user and group information in WebLogic Server's embedded LDAP server.

When using the DefaultAuthenticator, user passwords can be reset directly by accessing the Users and Groups tab of the security realm's settings in the WebLogic Admin Console.

Additionally, BDI Job Admin credentials can be added, deleted, updated and listed through a web service API exposed by the Job Admin BDI component. For more information, see *Managing Credentials Using curl in the Oracle® Retail Bulk Data Integration Implementation Guide*.

## Logging

Job Admin logs to multiple files, all located on the file system under the deployment path for the `ocds-batch-job-admin.war`. For more information, see *Monitoring Batch Jobs Using BDI Job Admin* in *Oracle® Retail Bulk Data Integration Implementation Guide* for additional details regarding Job Admin logging.

## URL patterns

OCDS Job Admin uses the following URL patterns.

**Table 3–5 URL Patterns**

Type	User
Web Application User Interface	<code>http://&lt;hostname&gt;:&lt;port&gt;/ocds-batch-job-admin/</code>
REST services	<code>http://&lt;hostname&gt;:&lt;port&gt;/ocds-batch-job-admin/ocds-batch-job-admin/resources</code>

## Job Admin User Interface

The OCDS BDI Job Admin UI is a web application that provides the GUI for managing batch jobs and runtime.

The User Interface provides ability to:

- Start/restart, and track status of jobs
- Trace data
- View diagnostic errors
- Manage options at job and system level
- View the logs

Job Admin is a standardized framework component used by Oracle Retail BDI Edge Applications. Detailed documentation for the component is available in the *Oracle® Retail Bulk Data Integration Implementation Guide*.



---



---

## RIB Injector

The OCDS (RIB) Injector is the interface between RIB infrastructure and OCDS; it listens for SOAP-based RIB messages for the families listed in the following table. The messages received by the OCDS Injector represent incremental changes to the merchandising data received in OCDS through BDI.

### RIB Families

**Table 4–1 RIB Families**

<b>Family</b>	<b>Message Type</b>
<i>DiffS</i>	DIFFCRE
	DIFFDEL
	DIFFMOD
<i>DiffGrp</i>	DIFFGRPHDRCRE
	DIFFGRPHDRMOD
	DIFFGRPDEL
	DIFFGRPDTLCRE
	DIFFGRPDTLDEL
	DIFFGRPDTLMOD
<i>ITEMLOC</i>	ITEMLOCCRE
	ITEMLOCMOD
	ITEMLOCDEL
	ITEMLOCREPLMOD

**Table 4–1 (Cont.) RIB Families**

<b>Family</b>	<b>Message Type</b>
<i>Items</i>	ITEMIMAGECRE
	ITEMIMAGEDEL
	ITEMIMAGEMOD
	ITEMCRE
	ITEMDEL
	ITEMHDRMOD
	RELITEMHEADCRE
	RELITEMHEADMOD
	RELITEMHEADDEL
	RELITEMDETCRE
	RELITEMDETMOD
	RELITEMDETDEL
	ITEMVATCRE
	ITEMVATMOD
	ITEMVATDEL
<i>MERCHHIER</i>	CLASSCRE
	CLASSMOD
	CLASSDEL
	DEPTCRE
	DEPTMOD
	DEPTDEL
	DIVISIONCRE
	DIVISIONMOD
	DIVISIONDEL
	GROUPCRE
	GROUPMOD
	GROUPDEL
	SUBCLASSCRE
	SUBCLASSMOD
	SUBCLASSDEL

**Table 4–1 (Cont.) RIB Families**

<b>Family</b>	<b>Message Type</b>
<i>ORGHIER</i>	ORGHIERCRE
	ORGHIERMOD
	ORGHIERDEL
	CHAINCRE
	CHAINMOD
	CHAINDEL
	AREACRE
	AREAMOD
	AREADEL
	REGIONCRE
	REGIONMOD
	REGIONDEL
	DISTRICTCRE
	DISTRICTMOD
	DISTRICTDEL
<i>STORES</i>	STORECRE
	STOREDEL
	STOREMOD
	STOREDTLCRE
	STOREDTLDEL
	STOREDTLMOD
<i>WH</i>	WHCRE
	WHDEL
	WHMOD
	WHDTLCRE
	WHDTLDEL
	WHDTLMOD

## Logging

The OCDS RIB Injector writes log messages to a log file on the file system under the deployment path for the `injector.war`. The log level is controlled by the `ocds-injector.war\WEB-INF\lib\injector.jar\log4j.xml` file.

## WebLogic Users and Groups

WebLogic security realms provide a mechanism for protecting WebLogic resources. Users are entities that can be authenticated in a security realm. Users are organized into Groups that can have different levels of access to WebLogic resources.

This section lists the default security realm's Groups and Users used by the RIB Injector component.

## Groups

**Table 4–2** *WebLogic Groups*

Name	Description	Component
IntegrationGroup	SOAP Authentication User Group	RIB Injector

## Users

**Table 4–3** *WebLogic Users*

Type	Group	Description	Component
Integration User	IntegrationGroup	RIB Web Service User	RIB Injector

## Changing User Passwords

The OCDS Injector's user password is authenticated by the default Authentication Provider configured for the security realm of the WebLogic domain. The WebLogic security architecture supports a variety of types of Authentication Providers, including custom. The out-of-the-box Authentication Provider is the WebLogic Authentication provider, also known as the DefaultAuthenticator, which accesses user and group information in WebLogic Server's embedded LDAP server.

When using the DefaultAuthenticator, a user password can be reset directly by accessing the Users and Groups tab of the security realm's settings in the WebLogic Admin Console.

---

---

## ORDS Web Services

The RESTful API enabling Omnichannel applications to access the Merchandising and Pricing data from OCDS runs on Oracle REST Data Services (ORDS).

### Performance Tuning

As the number of Omnichannel clients of OCDS increases, it is possible to experience performance degradation due to there being too few database connections to service the load. ORDS uses the Oracle Universal Connection Pool (UCP), a cache of reusable database connection objects. The out-of-the-box ORDS defaults for initial number of connections created and the maximum number of connections allowed is 3 and 10 respectively, which may be low for very active environments.

These properties, and others, are stored in the Oracle REST Data Services `<config-folder>/ords/defaults.xml` file that is created at the install time.

Increase these connection limits by setting the `jdbc.InitialLimit` and `jdbc.MaxLimit` configuration file parameters to higher values.

See section *B.4 Understanding the Configuration File Format* of the Oracle REST Data Services Documentation Release 18.1 (*Oracle REST Data Services Installation, Configuration, and Development Guide*) for details on all ORDS configurations.

### REST Services

This chapter deals with the REST Services.

#### Authentication

REST Services are secured with SSL and basic authentication. Successful authentication of a REST request requires the caller provide the username and password of a WebLogic User in the `OcdsMonitorGroup`.

#### Endpoints

The root URL pattern for the OCDS (ORDS) Web Services is shown below, where `{path-prefix}` is defined at install time.

```
http[s]://{host}[:{port}]/ords/{path-prefix}/omnichannel/
```

For individual URL endpoint patterns, see this guide's section titled: [Appendix A: REST Web Service API](#).

## WebLogic Users and Groups

WebLogic security realms provide a mechanism for protecting WebLogic resources. Users are entities that can be authenticated in a security realm. Users are organized into Groups that can have different levels of access to WebLogic resources.

This section lists the default security realm's Groups and Users used by the ORDS web service component.

### Groups

**Table 5-1 WebLogic Groups**

Name	Description	Component
OcdsMonitorGroup	REST Authentication User Group	ORDS

### Users

The credentials of Users in the OcdsMonitorGroup are used for REST web service authentication. It is recommended that one user be created for each Omnichannel application that is integrated with OCDS.

**Table 5-2 WebLogic Users**

Type	Group	Description	Component
Omnichannel Application User	OcdsMonitorGroup	Omnichannel Applications Web Service User	ORDS

## Changing User Passwords

The user password, supplied in HTTP requests to any OCDS (ORDS) Web Service endpoint, is authenticated by the default Authentication Provider configured for the security realm of the WebLogic domain. The WebLogic security architecture supports a variety of types of Authentication Providers, including custom. The out of the box Authentication Provider is the WebLogic Authentication provider, also known as the DefaultAuthenticator, which accesses user and group information in WebLogic Server's embedded LDAP server.

When using the DefaultAuthenticator, a user password can be reset directly by accessing the Users and Groups tab of the security realm's settings in the WebLogic Admin Console.

This chapter deals with data retention.

## Data Retention

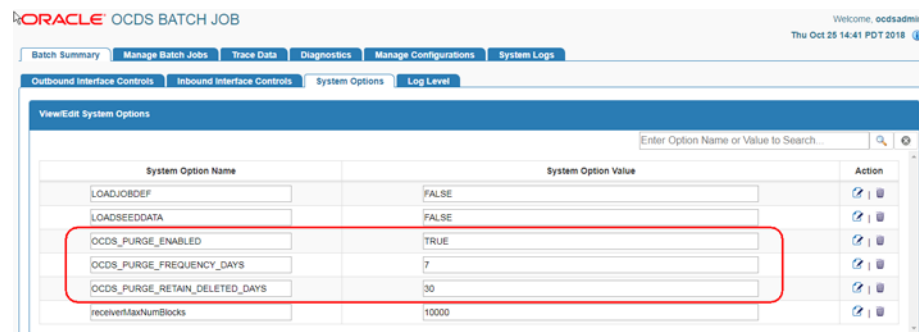
OCDS serves as a common, centralized, repository of the merchandising and pricing data required by Omnichannel applications. At any point, a downstream Omnichannel application should be able to ask OCDS for all data changes *since* a point in time. In addition, OCDS clients should be able to depend on OCDS to provide a complete snapshot of all Merchandising and Pricing data, if needed.

As Merchandising and Pricing data are added in the upstream source systems, records are added to OCDS's transactional table through BDI and/or RIB communication. When data has been deleted from the source system, OCDS continues to retain records with the state of the data marked as deleted so the "delete" event can be conveyed to OCDS clients.

## Purging Deleted Data

OCDS has a built-in mechanism to purge deleted data, at an interval, after the data has been flagged "deleted" for a certain number of days. The OCDS purge mechanism is enabled by default, and can be configured to a retailer's business needs from the OCDS Job Admin UI.

**Figure 6–1 OCDS Batch Job - System Options**



Use the OCDS Job Admin System Options to view and edit the OCDS Purge property values.

**Table 6–1 System Options**

<b>System Option Name</b>	<b>Description</b>
OCDS_PURGE_ENABLED	Indicates if deleted data should be purged.
OCDS_PURGE_RETAIN_ DELETED_DAYS	The number of days to retain deleted data. The number of days should be enough to be certain all Omnichannel applications have learned about any "deletes".
OCDS_PURGE_ FREQUENCY_DAYS	The frequency, described in days, with which deletes are to be purged.



# A

---

---

## Appendix A: REST Web Service API

This chapter describes the REST Web Service API

### diffs: Get Item Differentiators

Get the *changes* made to Diffs *since* a point in time. Diffs are special purpose attributes used to define concepts that differentiate items that are closely related, but not the same. Diffs are used to define colors, sizes, patterns, scents and flavors.

```
GET ords/{system}/omnichannel/{version}/diff
```

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp

Parameter	Value	Description	Optional	Parameter Type	Data Type
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.

Field	Description
diffid	The unique ID of the diff. Data will always be present in this field.
diffdesc	The type code for the diff. All diffs belong to one any only one type.
difftype	A value of the types of differentiators contained in this differentiator group, such as S - size, C - color, F - flavor, E - scent, P - pattern. Valid values are stored in the DIFF_TYPE table.
difftypedesc	The description of the diff. Data will always exist in this field.

## Example Data

**Content-Type:** application/json **Usage:**

- `https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/diff?since=2018-01-23T16:26:41.485Z:30&before=2018-06-23T16:26:41.485Z`

```
{
  "items": [
    {
      "action": "INSERT",
      "diffid": "AUTOBLACK",
      "diffdesc": "Auto Black",
      "difftype": "AUTO_C",
      "difftypedesc": "Auto Color"
    },
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1,
  "links": [ {.. } ]
}
```

## diff/group: Get Item Differentiator Group

Get the *changes* made to Diff Groups *since* a point in time.

GET `ords/{system}/omnichannel/{version}/diff/group`

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism where by callers can exclude data-changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
diffgroupid	A unique number id for the differentiator group. Data will always be present in this field.
diffgroupdesc	Description of the differentiator group (for example: Mens Shirt Sizes, Womens Shoe Sizes, Girls Dress Sizes, Shower Gel Scents, Yogurt Flavors, and so on). Description data is only sent in the primary integration language of the system.
difftypeid	A value of the types of differentiators contained in this differentiator group, including but not limited to: S (size), C (color), F (flavor), E (scent), P (pattern).
diffid	Contains the description of the differentiator type. For Example: Size, Color, Flavor, Scent, Pattern. Description data is only sent in the primary integration language of the system.

## Example Data

**Content-Type:** application/json**Usage:**

<https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/diff/group?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "diffgroupid": "AUTO CLRS",
      "diffgroupdesc": "Auto Basic Color Pallet",
      "difftypeid": "AUTO_C",
      "diffid": "1"
    }
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1,
  "links": [ { .. } ]
}
```

}

## item: Get Item

Get *changes* made to items *since* a point in time.

```
GET ords/{system}/omnichannel/{version}/item
```

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
nodelevel	<ul style="list-style-type: none"> <li>▪ COMPANY</li> <li>▪ CHAIN</li> <li>▪ AREA</li> <li>▪ REGION</li> <li>▪ DISTRICT</li> <li>▪ STORE</li> <li>▪ WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with nodeId to identify the node.	Yes	Query	String

Parameter	Value	Description	Optional	Parameter Type	Data Type
nodeid	0 .. n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
itemlocation	The Organization Hierarchy Node id when requested by node, otherwise "ent" (Enterprise) is returned.
item	ID of item. Data will always be present in this field.

<b>Field</b>	<b>Description</b>
itemparent	ID identifies the item/group at the level above the item. This value must exist as an item in another row on the item_master table.
itemgrandparent	ID identifies the item/group two levels above the item. This value must exist as both an item and an item parent in another row on the item_master table.
itemlevel	Number indicating which of the three levels the item resides. The item level determines if the item stands alone or if it is part of a family of related items. The concept is best explained with typical (although not exhaustive) examples. Staple items generally have a item level = 1 UPCs for Staple items generally have an item level = 2 (and the staple item will be the UPCs parent item). Fashion styles generally have an item level = 1. Fashion skus generally have an item level = 2. If UPCs for fashion skus generally have an item level = 3. Valid values are 1, 2 and 3. This field will always have data.



Field	Description
tranlevel	<p>Number indicating which of the three levels transactions occur for the items group. The transaction level is the level at which the items inventory is tracked in the system. The transaction level item will be inventoried, counted, transferred, shipped, and so on. Only one level of the hierarchy of an item family may contain transaction level items. The concept is best explained with typical (although not exhaustive) examples. Staple items generally have a TranLevel = 1. UPCs for Staple items generally have an TranLevel = 1 (inventory txns occur at the staple sku level; sales of the item roll up to the parent staple sku). Fashion styles generally have a TranLevel = 2 (the style itself is not sold/inventoried). Fashion skus generally have an TranLevel = 2 (the fashion sku is sold/inventoried). If UPCs for fashion skus generally have an TranLevel = 2 (the fashion sku is sold/inventoried). There are some rare cases in vendor managed inventory where the TranLevel = 3. Valid values are 1, 2 and 3. This field will always have data.</p>
inventoryind	<p>This indicator is used to determine if inventory is held for the item/item family. Inventory is held for most items. However, inventory is not held (value = N) in some special cases, such as: Concession items (items that are sold by independent in location concessions), Consignment items (items are not owned by the retailer; financial and inventory processing occurs after the item is sold to a consumer), Containers sold then returned for deposit and some items that are transformed for sale. Valid values are Y and N. This field will always have data.</p>

<b>Field</b>	<b>Description</b>
diff1level	This field will contain either ID or GROUP, based on whether the diff_1 is a group diff or a diff id.
diff1type	This field will hold a value of the types of differentiators contained in this differentiator group, such as S - size, C - color, F - flavor, E - scent, P - pattern. Valid values are stored in the DIFF_TYPE table.
diff1	Diff_group or diff_id that differentiates the current item from its item_parent. For an item that is a parent, this field may be either a group (that is Mens pant sizes) or a value (6 oz).
diff2level	This field will contain either ID or GROUP, based on whether the diff_2 is a group diff or a diff id.
diff2type	This field will hold a value of the types of differentiators contained in this differentiator group, such as S - size, C - color, F - flavor, E - scent, P - pattern. Valid values are stored in the DIFF_TYPE table.
diff2	Diff_group or diff_id that differentiates the current item from its item_parent. For an item that is a parent, this field may be either a group (that is Mens pant sizes) or a value (6 oz).
diff3level	This field will contain either ID or GROUP, based on whether the diff_3 is a group diff or a diff id.
diff3type	This field will hold a value of the types of differentiators contained in this differentiator group, such as S - size, C - color, F - flavor, E - scent, P - pattern. Valid values are stored in the DIFF_TYPE table.
diff3	Diff_group or diff_id that differentiates the current item from its item_parent. For an item that is a parent, this field may be either a group (that is Mens pant sizes) or a value (6 oz).

<b>Field</b>	<b>Description</b>
division	Number identifying the Division in the merchandise hierarchy to which the item belongs. If the item has a parent, the parent and item will belong to the same division. This data is required
groupid	Number identifying the Group in the merchandise hierarchy to which the item belongs. If the item has a parent, the parent and item will belong to the same group. This data is required
dept	Number identifying the department in the merchandise hierarchy to which the item belongs. If the item has a parent, the parent and item will belong to the same department. This data is required
class	Number identifying the class in the merchandise hierarchy to which the item belongs. If the item has a parent, the parent and item will belong to the same department and class. Class is not a unique ID (for example, every department can have a class 1). This data is required.
uniqueclass	Number identifying the class in the merchandise hierarchy to which the item belongs. If the item has a parent, the parent and item will belong to the same department and class. Class is not a unique ID (for example, every department can have a class 1). This data is required.
subclass	Number identifying the subclass in the merchandise hierarchy to which the item belongs. If the item has a parent, the parent and item will belong to the same department, class and subclass. Subclass is not a unique ID (for example, every department/class can have a subclass 1). This data is required.

<b>Field</b>	<b>Description</b>
uniquesubclass	Number uniquely identifying the subclass node which the item belongs to. Subclass is not unique ID the merchandise hierarchy. The combination of Dept/Class/Subclass is unique, but requires use of a composite key. The composite key is generally used in user interfaces. The unique ID can be used in back end processing or in systems that cannot have a composite key for a node in the merchandise hierarchy.
description	Primary description of the item in the integration language of the system. This value is required.
localdescription	The local description of the item, when requesting by Organization Hierarchy Node and Hierarchy level is "STORE". This may be the same as the primary description of the item, a regional description of the item (for example, jimmies vs. sprinkles in the US or roll vs. bap vs. cob vs. bun in the UK), or a value in a local language (for example, Overlay dress true black knit at US stores vs. Lagenkleid - Strick, tiefschwarz at stores in Germany). The intent is that this string is appropriate to print description on signage/receipts at this location. This will be null for all non-STORE Hierarchy levels.

<b>Field</b>	<b>Description</b>
merchandiseind	Indicates if the item is a merchandise item (Y, N). Merchandise items are generally physical items (things that must be shipped/received and of which there is an inventory). Non merchandise items are often items which do not have inventory. Common examples include extra fees for service (extended warranties, alterations) or endlessly available items (downloads, in app purchases of bonus content, subscriptions). All items, both merchandise and non-merchandise are exported from RMS. This value is required.
uintype	The unique identification number (UIN) used to identify the instances of the item at the location.
mfgrecretail	Manufacturers recommended retail price for the item. Used for informational purposes only. This field is stored in the primary currency.
originalunitretail	The original retail price of the item per unit. This field is stored in the primary currency.
catchweightind	Indicates whether the item should be weighed when it arrives at a location. Valid values for this field are Y and N. This field will always have data;
itemservicelevel	Holds a value that restricts the type of shipment methods that RCOM can select for an item.
giftwrapind	This field will contain a value of Y if the item is eligible to be gift wrapped. If not explicitly defined, this field will default to N. This field will always have data records.
shipaloneind	This field will contain a value of Y if the item must be shipped alone to consumers. If not explicitly defined, this field will default to N. This field will always have data records.

Field	Description
standarduom	Unit of measure in which stock and or financials of the item is tracked at a corporate level. Unit of measure reference values may have to be manually synced between the systems as this is foundation data that is not currently bulk integrated out of Merch. This value is required.
xdiff1desc	When diff1level is ID, the description of the diff. When diff1level is GROUP, the description of the differentiator group.
xdiff2desc	When diff2level is ID, the description of the diff. When diff2level is GROUP, the description of the differentiator group.
xdiff3desc	When diff3level is ID, the description of the diff. When diff3level is GROUP, the description of the differentiator group.

## Example Data

**Content-Type:** application/json  
**Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/location/retailstore?nodeId=1&nodeLevel=CHAIN&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "itemlocation": "ent",
      "item": "100805000",
      "itemparent": null,
      "itemgrandparent": null,
      "itemlevel": 1,
      "tranlevel": 2,
      "inventoryind": "Y",
      "diff1level": "GROUP",
      "diff1type": "AUTO_S",
      "diff1": "AUTO XS-XL",
      "diff2level": "GROUP",
      "diff2type": "C",
      "diff2": "BASIC CLRS",
      "diff3level": null,
      "diff3type": null,
      "diff3": null,
      "division": 2000,
      "groupid": 8000,
      "dept": 8787,
      "class": 16,
    }
  ]
}
```

```

"uniqueclass": 294,
"subclass": 16,
"uniquesubclass": 203,
"description": "DK Cala Lily",
"localdescription": null,
"merchandiseind": "Y",
"uintype": null,
"mfgrecretail": null,
"originalunitretail": 9.99,
"catchweightind": "N",
"itemservicelevel": null,
"giftwrapind": "N",
"shipaloneind": "N",
"standarduom": 1,
"xdiff1desc": "Auto XS-XL",
"xdiff2desc": "Basic Color Pallet",
"xdiff3desc": null}
],
"hasMore": true,
"limit": 1,
"offset": 0,
"count": 1,
"links": [ {.. } ]
}

```

## item/dimensiontype: Get Item Dimension Type

Get *changes* made to item dimension types *since* a point in time. Dimension Types are the types of dimensions associated with a dimension system. This information is derived from the current item diff usage.

GET ords/{system}/omnichannel/{version}/item/dimensiontype

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp

Parameter	Value	Description	Optional	Parameter Type	Data Type
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.



Field	Description
dimensionsystem	A system-defined code (Style ID) for the generalized dimension structure an item belongs to.
dimension	The specific code (Diff Type) values belonging to a dimension system. For example, COLOR, SIZE, and so on.
description	The text description for a dimension code (Diff Type) used for display.

## Example Data

**Content-Type:** application/json**Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/dimensiontype?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>
- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/dimensiontype?nodeId=5111&nodeLevel=STORE&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
    }
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1,
  "links": [ {.. } ]
}
```

## item/dimensionvalue: Get Item Dimension Value

Get *changes* made to item dimension values *since* a point in time. Dimension Values are the values associated with each dimension within a dimension system. This information is derived from the current item diff usage.

```
GET ords/{system}/omnichannel/{version}/item/dimensionvalue
```

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
dimensionsystem	A system-defined code (Style ID) for the generalized dimension structure an item belongs to.
dimension	The specific code (DiffType) values belonging to a dimension system. For example, COLOR, SIZE, and so on.
value	The specific value (Diff) that exists for a given dimension type for a given dimension system. This is a coded value, for example, S, M, L, and so on.
description	The text description of the value code (Diff).

## Example Data

**Content-Type:** application/json**Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/dimensionvalue?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>
- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/dimensionvalue?noDelId=5111&nodeLevel=STORE&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
    }
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1,
  "links": [ {.. } ]
}
```

## item/image: Get Item Image

Get *changes* made to item images *since* a point in time.

GET ords/{system}/omnichannel/{version}/item/image

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
nodelevel	<ul style="list-style-type: none"> <li>▪ COMPANY</li> <li>▪ CHAIN</li> <li>▪ AREA</li> <li>▪ REGION</li> <li>▪ DISTRICT</li> <li>▪ STORE</li> <li>▪ WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with nodeid to identify the node.	Yes	Query	String
nodeid	0..n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer

Parameter	Value	Description	Optional	Parameter Type	Data Type
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
item	Contains the unique alphanumeric identifier for the item, the image is for.
imagename	Contains the name of the image of the item
imageaddr	Contains the actual path where the file of the image of the item is stored.
imagedesc	Contains the type of the image of the item. Valid values are defined as member of IITD code type.
imagetype	Contains the type of the image of the item. Valid values are defined as member of IITD code type.

Field	Description
primaryind	Indicates whether this record is the primary image of the item or not. Valid values are Y(es) and N(o) only. Default to N value if left blank or set as NULL.
displaypriority	This field will specify the display sequence order of images associated to the item per priority.

## Example Data

**Content-Type:** application/json**Usage:**

- `https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/image?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z`
- `https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/image?nodeId=5111&nodeLevel=STORE&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z`

```
{
  "items": [
    {
      "action": "INSERT",
    }
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1,
  "links": [ {.. } ]
}
```

## item/itemlocation: Get Item Location

Get *changes* made to item locations *since* a point in time.

GET `ords/{system}/omnichannel/{version}/item/itemlocation`

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp

Parameter	Value	Description	Optional	Parameter Type	Data Type
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
nodelevel	<ul style="list-style-type: none"> <li>▪ COMPANY</li> <li>▪ CHAIN</li> <li>▪ AREA</li> <li>▪ REGION</li> <li>▪ DISTRICT</li> <li>▪ STORE</li> <li>▪ WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with nodeid to identify the node.	Yes	Query	String
nodeid	0..n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer
limit	0..n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer

Parameter	Value	Description	Optional	Parameter Type	Data Type
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
loc_type	Describes the type of location. Valid values include S (store) and W (warehouse).
location	Numeric ID of location. The intersection of location and item is a distinct entity. Data will always be present in this field.
item	ID of item. The intersection of location and item is a distinct entity. Data will always be present in this field.
sellingunitretail	The unit retail price in the selling unit of measure for the item/location combination. This field is stored in the local currency.
sellinguom	Contains the selling unit of measure for an items single-unit retail.
taxableind	Indicates if item is taxable at the store



<b>Field</b>	<b>Description</b>
localitemdesc	Contains the local description of the item. This may be the same as the primary description of the item, a regional description of the item (for example, jimmies vs. sprinkles in the US or roll vs. bap vs. cob vs. bun in the UK), or a value in a local language (for example, Overlay dress true black knit at US stores vs. Lagenkleid - Strick, tiefschwarz at stores in Germany). The intent is that this string is appropriate to print description on signage/receipts at this location.
status	Current status of item at the store.
qtykeyoptions	Determines whether the quantity key on a POS should be used for this item at the location.
manualpriceentry	Determines whether the price can/should be entered manually on a POS for this item at the location.
foodstampind	Indicates whether the item is approved for food stamps at the location. This value will be downloaded to the POS.
fixedtarevalue	Holds the value associated of the packaging in items sold by weight at the location.
fixedtareuom	Holds the unit of measure value associated with the tare value. The only processing RMS does involving the fixed tare value and UOM is downloading it to the POS.
stopsaleind	Indicates that sale of the item should be stopped immediately at the location (that is in case of recall and so on).
returnableind	Indicates if the item can be returned to the location
backorderind	Indicates if the item can be back ordered to the location.
merchandiseind	Indicates if the item is a merchandise item (Y, N).

Field	Description
clearanceind	Indicates if item is on clearance at the store.
crosssell	Indicates whether the item is a related item with either a "UPSL" or CRSL relationship type value.
attacheditem	Indicates whether the item is a related item with either a "SUBS" relationship type value.
varcodes	List of the applicable vat codes and active dates

## Example Data

**Content-Type:** application/json **Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/itemlocation?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>
- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/itemlocation?nodeId=5111&nodeLevel=STORE&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
"items": [
  {
    "action": 'INSERT',
    "loc_type": 'S',
    "location": 1151,
    "item": '100000008',
    "sellingunitretail": 12,
    "sellinguom": 'EA',
    "taxableind": 'N',
    "localitemdesc": 'Test Item 100000008',
    "status": 'A',
    "qtykeyoptions": null,
    "manualpriceentry": null,
    "foodstampind": null,
    "fixedtarevalue": null,
    "fixedtareuom": null,
    "stopsaleind": null,
    "returnableind": null,
    "backorderind": 'N',
    "merchandiseind": 'Y',
    "clearanceind": 'N' "crosssell": 'N',
    "attacheditem": 'N',
    "vatcodes":
      [ {'code': 'AAA', 'date': '2018-01-01'}, {'code': 'AAB', 'date': '2018-03-01'}, {'code': 'AAC', 'date': '2018-06-01'}, {'code': 'AAD', 'date': '2018-09-01'} ]
  }
],
"hasMore": true,
"limit": 1,
"offset": 0,
"count": 1,
"links": [ {.. } ]}
```

## item/price: Get Item Price

Get *changes* made to item prices *since* a point in time.

GET `ords/{system}/omnichannel/{version}/item/price`

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
nodelevel	<ul style="list-style-type: none"> <li>▪ COMPANY</li> <li>▪ CHAIN</li> <li>▪ AREA</li> <li>▪ REGION</li> <li>▪ DISTRICT</li> <li>▪ STORE</li> <li>▪ WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with <code>nodeId</code> to identify the node.	Yes	Query	String

Parameter	Value	Description	Optional	Parameter Type	Data Type
nodeid	0 .. n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer
priceType	INITIAL REGULAR CLEARANCE	Request data for a single Price Type	Yes	Query	Integer
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
location	Location identifier for item price
loctype	Location type
item	Item identifier

Field	Description
pricetype	INITIAL, REGULAR, CLEARANCE
price	item price
effective	Date time when change becomes effective
eventid	Pricing Service event id for the price activity
resetind	Flag to indicate to reset the item to previous price

## Example Data

**Content-Type:** application/json **Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/price?since=2018-01-23T16:26:41.485+05:30&before=2018-06-23T16:26:41.485+05:30>
- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/price?since=2018-01-23T16:26:41.485+05:30&before=2018-06-23T16:26:41.485+05:30&pricetype=INITIAL>

```
"items": [
  o{
    "action": "INSERT",
    "location": 1151,
    "loctype": "S",
    "item": "100000008",
    "pricetype": "CLEARANCE",
    "price": 4.63,
    "effective": "2018-08-05T00:00:00",
    "eventid": 2000,
    "resetind": 1
  },
o],
"hasMore": false,
"limit": 10000,
"offset": 0,
"count": 5,
"links": [ {.. } ]}
```

## item/relateditem: Get Related Item

Get *changes* made to related items *since* a point in time.

GET ords/{system}/omnichannel/{version}/item/relateditem

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
nodelevel	<ul style="list-style-type: none"> <li>▪ COMPANY</li> <li>▪ CHAIN</li> <li>▪ AREA</li> <li>▪ REGION</li> <li>▪ DISTRICT</li> <li>▪ STORE</li> <li>▪ WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with nodeid to identify the Organization Hierarchy node.	Yes	Query	String
nodeid	0 .. n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer

Parameter	Value	Description	Optional	Parameter Type	Data Type
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how related item relationship has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.  <i>Important:</i> This response has two levels of actions. This field only communicates the action of the relationship.
relationshipid	Unique identifier for each relationship. Data will always exist in this field.
item	Item for which the relationships are defined. This is the parent item in a related item relationship. Data will always exist in this field.

Field	Description
mandatoryind	Indicates whether the relationship should be mandatory. For example, an item like a laptop may have a mandatory cross sell relationship. The related items could be different power cords for the US, UK, Mainland Europe, India, and so on. When the laptop is sold, it should be mandatory that one of the related power cords also be selected. Generally, only cross sell relationships are mandatory. Substitution and upsell relationships can be defined as mandatory, but in those cases, the definition of mandatory is at the discretion of the client and generally means that substitution or upsell must, as business process, be offered to consumers.
relationshiptype	Describes the type of relationship. Valid values include: CRSL (Cross Sell), SUBS (Substitution), UPSL (Up-sell).
relateditem	Item id of the related item. This is the item that should be Cross Sold, Substituted, or Up Sold when the item on the parent record is sold.
relateditemaction	Recommended action based on how the "related" item, in the related item relationship, has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.  <i>Important:</i> This response has two levels of actions. This field communicates the action of the "relateditem" involved in the related item relationship.
startdate	From this date related item can be used on transactions.
enddate	Till this date related item can be used on transactions. A value of null means that it is effective forever.



Field	Description
priority	Applicable only in case of relationship type SUBS. In case of multiple related substitute items, this column could be used (optional) to define relative priority.

## Example Data

**Content-Type:** application/json  
**Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item/relateditem?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "relationshipid": 1,
      "item": "101500004",
      "mandatoryind": "N",
      "relationshiptype": "SUBS",
      "relateditem": "100600003",
      "startdate": "2014-12-17",
      "enddate": "2014-12-31",
      "priority": 1
    },
    {
      "action": "INSERT",
      "relationshipid": 402,
      "item": "101500012",
      "mandatoryind": "N",
      "relationshiptype": "CRSL",
      "relateditem": "101500004",
      "relateditemaction": "DELETE",
      "startdate": null,
      "enddate": null,
      "priority": null
    },
    {
      "action": "INSERT",
      "relationshipid": 402,
      "item": "101500012",
      "mandatoryind": "N",
      "relationshiptype": "CRSL",
      "relateditem": "101500021",
      "relateditemaction": "INSERT",
      "startdate": null,
      "enddate": null,
      "priority": null
    }
  ],
  "hasMore": true,
  "limit": 3,
  "offset": 0,
  "count": 3,
  "links": [ {.. } ]
}
```

## location/retailstore: Get Retail Store

Get *changes* made to retail stores *since* a point in time. Address fields are always derived from the primary Business (addr\_type: 01) address.

```
GET ords/{system}/omnichannel/{version}/location/retailstore
```

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
nodelevel	<ul style="list-style-type: none"> <li>▪ COMPANY</li> <li>▪ CHAIN</li> <li>▪ AREA</li> <li>▪ REGION</li> <li>▪ DISTRICT</li> <li>▪ STORE</li> <li>▪ WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with nodeId to identify the node.	Yes	Query	String

Parameter	Value	Description	Optional	Parameter Type	Data Type
nodeid	0 .. n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
storeid	Unique ID of the store.
storename	Name of the store which, along with the store number, identifies the store.
storename10	A ten character abbreviation of the store name.
manager	Name of the store manager.
contactname	Name of the contact at this address

Field	Description
phonenumber	Phone number for the store.
faxnumber	Fax number for the store.
email	Email address for the store.
totalsqfeet	Total square footage of the store.
sellingsqfeet	Total square footage of the stores selling area.
currencycode	Currency code under which the store operates.
vatregion	Indicates whether or not Value Added Tax will be included in the retail prices for the store. Valid values are Y or N.
address1	First line of the Primary Business address. When the store has a primary business address, this field will contain data.
address2	Second line of the address.
address3	Third line of the address.
city	Name of the city that is associated with the address. When the store has a primary business address, this field will contain data.
state	The state abbreviation for the address.
country	The ISO 3166-1 country code associated with the address. When the store has a primary business address, this field will contain data.
postalcode	Zip code for the address.

## Example Data

**Content-Type:** application/json **Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/location/retailstore?nodeId=1&nodeLevel=CHAIN&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "storeid": 2689,
      "storename": "RMS 15.0 DTI Store",
      "storename10": "DTI STORE",
      "manager": "Mr. Jones",
      "contactname": "Sally Smith" "phonenumber": "123456",
      "faxnumber": "123456",
    }
  ]
}
```

```

"email": "rmsdti@gmail.com",
"totalsqfeet":20000
"sellingsqfeet":18000
"currencycode": "USD",
"vatregion": 1000,
"address1": "addr",
"address2": null,
"address3": null,
"city": "mpls",
"state": "MN",
"country": "US",
"postalcode": "55345"
}
],
"hasMore": true,
"limit": 1,
"offset": 0,
"count": 1,
"links": [ {.. } ]
}

```

## location/warehouse: Get Warehouse

Get *changes* made to warehouses *since* a point in time. Address fields are always derived from the primary Business (addr\_type: 01) address.

GET ords/{system}/omnichannel/{version}/location/warehouse

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp

Parameter	Value	Description	Optional	Parameter Type	Data Type
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.

<b>Field</b>	<b>Description</b>
warehouseid	Number which uniquely identifies the warehouse. The wh table stores all warehouses in the system. Both virtual and physical warehouses will be stored on this table. The addition of the new column, physical_wh, helps determine which warehouses are physical and which are virtual. All physical warehouses will have a physical_wh column value equal to their wh number. Virtual warehouses will have a valid physical warehouse in this column.
warehousename	Name of the warehouse which, along with the warehouse number, identifies the warehouse.
whnamesecondary	Secondary name of the warehouse.
contactname	Name of the contact at this address
phonenumber	Phone number of the contact person
faxnumber	Fax number of the contact person
email	Email address of the partner or suppliers representative contact.
currencycode	Currency code under which the warehouse operates.
physicalwh	The number of the physical warehouse that is assigned to the virtual warehouse.
vatregion	Vat Region in which warehouse is located
orghiertype	Organization type that will be used in reporting purposes for the warehouse. The type comes from the organizational hierarchy. Valid values are: 1 = Company 10 = Chain 20 = Area 30 = Region 40 = District 50 = Store
orghiervalue	Code associated with the specific organizational hierarchy type. Valid values include the company number, chain number, area number, and so on.

Field	Description
channelid	Channel for which the virtual warehouse will be assigned.
channelname	Name of the channel.
address1	Primary Business address. When the store has a primary business address, this field will contain data.
address2	Second line of the address.
address3	Third line of the address.
city	Name of the city that is associated with the address. When the store has a primary business address, this field will contain data.
state	State abbreviation for the address.
country	The ISO 3166-1 country code associated with the address. When the store has a primary business address, this field will contain data.
postalcode	Zip code for the address.

## Example Data

**Content-Type:** application/json **Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/location/warehouse?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [],
  "hasMore": true,
  "limit": 10000,
  "offset": 0,
  "count": 0,
  "links": [ {.. } ]
}
```

## merchhier: Get Merchandise Hierarchy

Get *changes* made to the merchandise hierarchy *since* a point in time. The merchandise hierarchy enables the grouping of items by Division, Group, Department, Class and Subclass.

```
GET ords/{system}/omnichannel/{version}/merchhier
```



## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0..n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer

Parameter	Value	Description	Optional	Parameter Type	Data Type
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
nodeid	This information identifies the node of the merchandise hierarchy that is described by this record. This field can not be null. HierarchyNodeId is only unique within an HierarchyLevel (meaning it is possible, for example, that there is both a DIVISION 1 and a GROUP 1 in the full merchandise hierarchy).
hierarchylevel	Name of the merchandise hierarchy entity. Description data is only sent in the primary integration language of the system.
nodename	Level of the merchandise hierarchy above the current node. Both ParentLevel and ParentId are should be evaluated to correctly traverse the hierarchy
parentlevel	Level of the merchandise hierarchy above the current node. Both ParentLevel and ParentId are should be evaluated to correctly traverse the hierarchy

Field	Description
parentnodeid	Id of the level of the merchandise hierarchy above the current node. Both ParentLevel and ParentNodeId should be evaluated to correctly traverse the hierarchy.
merchdisplayid	Holds the merchandise hierarchy id for division, group, department, class, and subclass.

## Example Data

**Content-Type:** application/json **Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/merchhier?since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>
- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/item?nodeId=5111&nodeLevel=STORE&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "nodeid": 4,
      "hierarchylevel": "SUBCLASS",
      "nodename": "Mens RMS 15.0 DTI",
      "parentlevel": "CLASS",
      "parentnodeid": 4
      "merchdisplayid":400
    }
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 60,
  "count": 1,
  "links": [ {.. } ]
}
```

## metadata-catalog: Get API Catalog

Get available resources in the specified versions of the API.

GET ords/{system}/omnichannel/{version}/metadata-catalog

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
limit	Number	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	Number	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
name	resource name
links	NA
ref	link relation
href	hypertext reference
mediaType	response media type

## Example Data

**Content-Type:** application/json

**Usage:** https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/metadata-catalog

```
{
  "items": [
    {
      "name": "v1",
      "links": [
        {
          "rel": "describes",
          "href": "http://10.135.189.123:8888/ords/{path-prefix}/omnichannel/v1/diff"
        },
        {
          "rel": "canonical",
          "href":

```

```

"http://10.135.189.123:8888/ords/{path-prefix}/omnichannel/metadata-catalog/v1/dif
f",
"mediaType": "application/json"
},
{
"rel": "alternate",
"href":
"http://10.135.189.123:8888/ords/{path-prefix}/omnichannel/open-api-catalog/v1/",
"mediaType": "application/openapi+json"
}
]
}
],
"hasMore": true,
"limit": 1,
"offset": 0,
"count": 1,
"links": [ {.. } ]
}

```

## orghier: Get Organization Hierarchy

Get *changes* made to the organizational hierarchy *since* a point in time. The organizational hierarchy describes the operational structure of a company

GET ords/{system}/omnichannel/{version}/orghier

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp

Parameter	Value	Description	Optional	Parameter Type	Data Type
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
nodelevel	<ul style="list-style-type: none"> <li>▪ COMPANY</li> <li>▪ CHAIN</li> <li>▪ AREA</li> <li>▪ REGION</li> <li>▪ DISTRICT</li> <li>▪ STORE</li> <li>▪ WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with nodeid to identify the node.	Yes	Query	String
nodeid	0 .. n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer

Parameter	Value	Description	Optional	Parameter Type	Data Type
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
hierarchylevel	This information identifies the level of the organizational hierarchy that is described by this record. Value is always COMPANY, CHAIN, AREA, REGION, DISTRICT, STORE or WAREHOUSE. This field can not be null.
hierarchynodeid	This information identifies the node of the organizational hierarchy that is described by this record. This field can not be null. HierarchyNodeId is only unique within an HierarchyLevel (meaning it is possible, for example, that there is both a DISTRICT 1 and a REGION 1 in the full organizational hierarchy).
hierarchynodename	Name of the organizational hierarchy entity. This field can not be null. Description data is only sent in the primary integration language of the system.
parentlevel	This information identifies the parent level of the organizational hierarchy that is described by this record'

Field	Description
parentid	This information identifies the id of parent organizational hierarchy.
mgrname	NA

## Example Data

**Content-Type:** application/json **Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/orghier?nodeId=1&nodeLevel=COMPANY&since=2018-01-23T16:26:41.485Z&before=2018-06-23T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "hierarchylevel": "AREA",
      "hierarchynodeid": 1,
      "hierarchynodename": "USA*",
      "parentlevel": "CHAIN",
      "parentid": 1,
      "mgrname": null
    }
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1,
  "links": [ {.. } ]
}
```

## orghier/descendant: Get Organization Hierarchy Node Descendant

Get entire branch of the organizational hierarchy beginning with a specified node. This resource returns the latest state of the organizational hierarchy.

GET ords/{system}/omnichannel/{version}/orghier/descendant

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
nodelevel	<ul style="list-style-type: none"> <li>COMPANY</li> <li>CHAIN</li> <li>AREA</li> <li>REGION</li> <li>DISTRICT</li> <li>STORE</li> <li>WAREHOUSE</li> </ul>	Request data for a single node in the Organization Hierarchy. Must be used with nodeid to identify the node.	Yes	Query	String



Parameter	Value	Description	Optional	Parameter Type	Data Type
nodeid	0 .. n	Request data for a single node in the Organization Hierarchy. Must be used with nodelevel to identify the node.	Yes	Query	Integer
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
hierarchylevel	This information identifies the level of the organizational hierarchy that is described by this record. Value is always COMPANY, CHAIN, AREA, REGION, DISTRICT, STORE or WAREHOUSE. This field can not be null.

Field	Description
hierarchynodeid	This information identifies the node of the organizational hierarchy that is described by this record. This field can not be null. HierarchyNodeid is only unique within an HierarchyLevel (meaning it is possible, for example, that there is both a DISTRICT 1 and a REGION 1 in the full organizational hierarchy).
hierarchynodename	Name of the organizational hierarchy entity. This field can not be null. Description data is only sent in the primary integration language of the system.
parentlevel	This information identifies the parent level of the organizational hierarchy that is described by this record'
parentid	This information identifies the id of parent organizational hierarchy.

## Example Data

**Content-Type:** application/json  
**Usage:**

- <https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/orghier/descendant?nodeId=1&nodeLevel=COMPANY>

```
{
  "items": [
    {
      "hierarchylevel": "CHAIN",
      "hierarchynodeid": 1,
      "hierarchynodename": "Chain 1*",
      "parentlevel": "COMPANY",
      "parentid": 1
    },
    {
      "hierarchylevel": "AREA",
      "hierarchynodeid": 1,
      "hierarchynodename": "USA*",
      "parentlevel": "CHAIN",
      "parentid": 1
    },
    {
      "hierarchylevel": "AREA",
      "hierarchynodeid": 2,
      "hierarchynodename": "Europe",
      "parentlevel": "CHAIN",
      "parentid": 1
    }
  ]
}
```

```

],
"hasMore": true,
"limit": 3,
"offset": 0,
"count": 3,
"links": [ {.. } ]
}

```

## refreshdate: Get Data Refresh Date

Get the date bulk data arrived into the system. The refreshdate can be compared to the SINCE request parameter, in REST calls to foundation data resources, to determine if response data represents "initial load" or an "incremental change".

GET `ords/{system}/omnichannel/{version}/refreshdate`

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
entitytype	<ul style="list-style-type: none"> <li>▪ diff</li> <li>▪ diffgroup</li> <li>▪ dimensiontype</li> <li>▪ dimensionvalue</li> <li>▪ item</li> <li>▪ itemimage</li> <li>▪ itemlocation</li> <li>▪ itemprice</li> <li>▪ merchhier</li> <li>▪ orghier</li> <li>▪ retailstore</li> <li>▪ warehouse</li> </ul>	Request refresh date for a single entity. This parameter is optional; when not used, refreshdates for all entities are returned.	Yes	Query	Integer
limit	Number	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer

Parameter	Value	Description	Optional	Parameter Type	Data Type
offset	Number	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
entity	The system entity. Each entity value correlates with one of the foundation data REST resources provided by this API
refreshdate	The timestamp data, for an entity, was initially loaded into system.

## Example Data

**Content-Type:** application/json

**Usage:**

- `https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/refreshdate`
- `https://{host}:{port}/ords/{path-prefix}/omnichannel/v1/refreshdate?entitytype=merchier`

```
{
  "items": [
    {
      "entity": "Item",
      "refreshdate": "2018-06-05T19:12:26.106Z"
    }
  ],
  "hasMore": false,
  "limit": 10000,
  "offset": 0,
  "count": 1,
  "links": [ { .. } ]
}
```

## inventory/future: Get Future Inventory

Get the *changes* made to Future Inventory positions *since* a point in time.

```
GET ords/{system}/omnichannel/{version}/inventory/future
```

## Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
item	A sellable and orderable transaction level item whose future inventory position is described. The item must be backorderable.
onorderquantity	Open order quantity for the item at all locations that are flagged as customer order locations and backorderable for the item. It is calculated as ordered quantity - allocated quantity for allocations to non-customer order locations.
receivedquantity	Received order quantity for the item at all locations.
backorderquantity	Back order quantity for the item at all locations that are flagged as customer order locations and backorderable for the item.

## Example Data

**Content-Type:** application/json

**Usage:**

- <https://{host}:{port}/ords/ocds/omnichannel/v1/inventory/future?since=2019-01-23T16:26:41.485Z:30&before=2019-07-01T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "item": "100400001",
      "onorderquantity": 50,
      "receivedquantity": 0,
      "backorderquantity": 5
    },
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1, "links": [ { .. } ]
}
```

## inventory/store: Get Store Inventory

Get the *changes* made to Available Store Inventory *since* a point in time.

GET `ords/{system}/omnichannel/{version}/inventory/store`

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer

Parameter	Value	Description	Optional	Parameter Type	Data Type
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.
item	Item whose current inventory position is described.
location	Location at which the item has inventory.
loctype	Type of location. S - store.
availablequantity	Available quantity of the item at the location. This quantity is calculated by subtracting transfer reserved quantity, customer reserved quantity, non sellable inventory and RTV from stock on hand (for example, current available inventory = stock on hand - (tsf reserved + customer reserved + non sellable + RTV qty)). Available inventory is in the standard unit of measure. For supplier records, -1 will be selected to indicate infinite inventory.
stockonhand	Current stock on hand for the item.
standarduom	Unit of measure in which stock of the item is tracked at a corporate level.

## Example Data

**Content-Type:** application/json



**Usage:**

- <https://{host}:{port}/ords/ocds/omnichannel/v1/inventory/store?since=2019-01-23T16:26:41.485Z:30&before=2019-07-01T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "item": "100850555",
      "location": "4241",
      "loctype": "S",
      "availablequantity": 5953,
      "stockonhand": 5978,
      "standarduom": "EA"
    },
  ],
  "hasMore": true,
  "limit": 1,
  "offset": 0,
  "count": 1, "links": [ {.. } ]
}
```

## inventory/warehouse: Get Warehouse Inventory

Get the *changes* made to Warehouse Inventory *since* a point in time.

GET ords/{system}/omnichannel/{version}/inventory/warehouse

### Parameters

Parameter	Value	Description	Optional	Parameter Type	Data Type
since	UTC ISO-8601 Date and Time	Request data changed since a specific point in time. Use is optional, if not used then changes since 00:00:00 (UTC) on January 1, 1970 will be returned.	Yes	Query	Timestamp

Parameter	Value	Description	Optional	Parameter Type	Data Type
before	UTC ISO-8601 Date and Time	Request data changed before a specific point in time. This parameter provides a mechanism whereby callers can exclude data changes occurring at the present time to avoid getting partial changes due to in-flight changes.	Yes	Query	Timestamp
limit	0 .. n	Limit response size to a fixed number of data items to control paging of result data. Use is optional. If not used then a default limit is used.	Yes	Query	Integer
offset	0 .. n	Request paged data beginning at a specific offset. Use is optional. If not used then entire first page of data is returned.	Yes	Query	Integer

## Response (items)

Field	Description
action	Recommended action based on how data has changed since a point in time. Possible values are: NO_CHANGE, DELETE, UPSERT, INSERT, UPDATE and NA.

Field	Description
item	Item whose current inventory position is described.
location	Warehouse at which the item has inventory. This is always a virtual warehouse.
loctype	Type of location. W represents the virtual warehouse.
availablequantity	Available quantity of the item at the location. This quantity is calculated by subtracting transfer reserved quantity, customer reserved quantity, non sellable inventory and RTV from stock on hand (e.g. current available inventory = stock on hand - (tsf reserved + customer reserved + non sellable + RTV qty)). Available inventory is in the standard unit of measure.
stockonhand	Current stock on hand for the item.
standarduom	Unit of measure in which stock of the item is tracked at a corporate level.
physicalwarehouse	This column will contain the number of the physical warehouse that is assigned to the virtual warehouse. It will only contain a value when the location is a warehouse.

## Example Data

**Content-Type:** application/json

**Usage:**

- <https://{host}:{port}/ords/ocds/omnichannel/v1/inventory/warehouse?since=2019-01-23T16:26:41.485Z:30&before=2019-07-01T16:26:41.485Z>

```
{
  "items": [
    {
      "action": "INSERT",
      "item": "101200191",
      "location": "141",
      "loctype": "W",
      "availablequantity": 0,
      "stockonhand": 0,
      "standarduom": "EA", "physicalwarehouse": 140
    }
  ],
}
```

```
"hasMore": true,  
"limit": 1,  
"offset": 0,  
"count": 1, "links": [ {.. } ]  
}
```