

webMethods Adapter for Apache HBase Installation and User's Guide

Version 9.8

August 2016

This document applies to webMethods Adapter for Apache HBase 9.8 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2016-2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <https://softwareag.com/licenses/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

Document ID: ADAPTER-WHB-IUG-98-20221202

Table of Contents

About this Guide.....	5
Document Conventions.....	6
Online Information and Support.....	7
Data Protection.....	8
1 Overview of the Adapter.....	9
About the Adapter.....	10
Architecture and Components.....	10
Package Management.....	11
Adapter Connections.....	12
Adapter Services.....	13
Viewing the Adapter's Update Level.....	17
Controlling Pagination.....	17
2 Installing and Uninstalling Adapter for Apache HBase.....	19
Overview of installing and uninstalling Adapter for Apache HBase.....	20
Requirements.....	20
The Integration Server Home Directory.....	20
Installing Adapter for Apache HBase.....	20
Uninstalling Adapter for Apache HBase.....	21
Migrating Retrieve Operation to Retrieve Operation With Comparator Service.....	22
3 Package Management.....	23
Overview of Package Management.....	24
Adapter for Apache HBase Package Management.....	24
Group Access Control.....	27
Adapter for Apache HBase in a Clustered Environment.....	27
4 Adapter for Apache HBase Connections.....	31
Overview of Adapter for Apache HBase Connections.....	32
Before Configuring or Managing Adapter Connection.....	32
Configuring an Adapter for Apache HBase Connection.....	32
Configuring an Adapter for Apache HBase Kerberos Connection.....	35
5 Adapter Services.....	37
Overview of Adapter Services.....	38
Before Configuring or Managing Adapter Services.....	38
Configuring Retrieve Operation Service.....	38
Configuring Delete Operation Service.....	44
Configuring Upsert Operation Service.....	46
Configuring Retrieve Operation With Comparator Service.....	49

6 Predefined Health Indicator.....	57
Predefined Health Indicator.....	58
7 Administrator APIs.....	59
Administrator APIs.....	60
8 Configuration Variables Templates for Adapter Assets in Microservices Runtime.....	61
Configuration Variables Templates for Adapter Assets in Microservices Runtime.....	62
9 Logging and Exception Handling.....	63
Overview of Logging and Exception Handling.....	64
Adapter for Apache HBase Logging Levels.....	64
Accessing Adapter Logging Levels.....	64
Changing Logging Levels.....	64
Adapter for Apache HBase Message Logging.....	64
Adapter for Apache HBase Error Codes.....	65

About this Guide

■ Document Conventions	6
■ Online Information and Support	7
■ Data Protection	8

This guide describes how to install, configure, and use the webMethods Adapter for Apache HBase. This guide also describes the built-in services provided by the Adapter for Apache HBase. It contains information for administrators and application developers who want to interact with Apache HBase through the use of Apache HBase client API.

To use this guide effectively, you should be familiar with:

- The basic concepts and tasks for working with Apache HBase
- The terminology and the basic operations of your operating system
- The setup and operation of webMethods Integration Server
- The basic concepts and tasks of Software AG Designer

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.softwareag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://techcommunity.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/u/softwareag> and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Overview of the Adapter

■ About the Adapter	10
■ Architecture and Components	10
■ Package Management	11
■ Adapter Connections	12
■ Adapter Services	13
■ Viewing the Adapter's Update Level	17
■ Controlling Pagination	17

About the Adapter

webMethods Adapter for Apache HBase is an add-on to webMethods Integration Server that enables you to interact with Apache HBase through the use of Apache HBase client API.

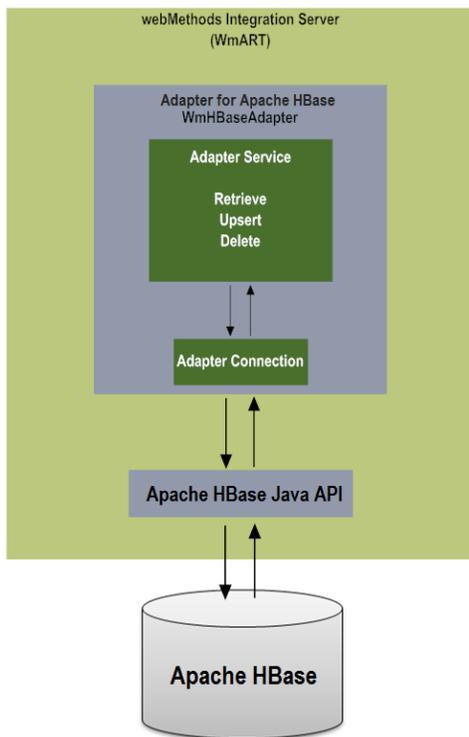
Using Adapter for Apache HBase, you can create and run adapter services to perform operations, such as retrieve, upsert and delete data from HBase tables.

For example, you can use Adapter for Apache HBase to easily retrieve or update particular set of records from a very large table, which is stored in different regional servers.

Architecture and Components

Adapter for Apache HBase provides a set of user interfaces, and services that enables you to create integration with HBase. The adapter is provided as a single package that must be installed on Integration Server. For detailed installation instructions, see [“Overview of installing and uninstalling Adapter for Apache HBase” on page 20](#). For software requirements, see *webMethods Adapters System Requirements*.

The following diagram shows at a high level, how the adapter components connect to HBase's backend.



- **webMethods Integration Server.** Adapter for Apache HBase is installed and runs on Integration Server.
- **(WmART).** The WmART package provides a common framework for webMethods Adapter for Apache HBase version 9.8 and later to use Integration Server's functionality, making

Integration Server the run-time environment for Adapter for Apache HBase. The WmART package is installed with Integration Server and it provides logging, error handling for the adapter, connections, and services.

- **Adapter for Apache HBase.** The Adapter for Apache HBase is delivered as a single package called WmHBaseAdapter. The adapter installation includes templates from which all adapter connections, and adapter services can be created. The adapter provides:
 - Integration Server Administrator user interfaces that will enable you to configure and manage adapter connections.
 - Software AG Designer user interfaces that will enable you to configure and manage adapter services.
- **Adapter services.** Adapter services enable the Integration Server to initiate and perform database operations on HBase. You configure adapter services using adapter services templates, which are provided with Adapter for Apache HBase. For more information about adapter services, see [“Overview of Adapter Services” on page 38](#).
- **Adapter connections.** Adapter connections enable the Integration Server to connect to HBase at runtime. You must configure an adapter connection before you can configure adapter services. For a detailed description of adapter connections, see [“Overview of Adapter for Apache HBase Connections” on page 32](#).
- **Apache HBase Java API.** Apache HBase java API is a client jar provided by HBase that connects to the HBase cluster.
- **Apache HBase.** Apache HBase is an open source, non-relational, column-oriented, distributed database that runs on top of HDFS(Hadoop Distributed Filesystem). It provides a fault-tolerant way of storing large collection of sparse data and is suited for faster read or write operations on large datasets with high throughput and low input/output latency.

Package Management

Adapter for Apache HBase is available as a package called WmHBaseAdapter. You can manage the WmHBaseAdapter package like any package on Integration Server

You can set up and manage the packages on Integration Server using the following considerations:

- Create user-defined packages for your connections and adapter services. For details, see [“Adapter for Apache HBase Package Management” on page 24](#).
- Understand how package dependencies work so that you can decide on how to manage your adapter services. For details, see [“Package Dependency Requirements and Guidelines” on page 24](#).
- You must have a control on which development groups have access to which adapter services. For details, see [“Group Access Control” on page 27](#).

Adapter Connections

Adapter for Apache HBase connects to HBase through Java API at run time. You create one or more connections at design time to use in integrations. The number of connections you create and the type of those connections, depend on the types of databases you are connecting to and your integration needs.

Adapter for Apache HBase connections contain parameters that Integration Server uses to manage connections on the Apache HBase. The adapter uses these connections to provide services. You configure connections using Integration Server Administrator. You must have Integration Server Administrator privileges to access Adapter for Apache HBase's administrative screens.

For instructions on configuring Adapter for Apache HBase connections, see [“Overview of Adapter for Apache HBase Connections” on page 32](#). For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.

Connection Pools

Integration Server includes a connection management service that dynamically manages connections and connection pools based on configuration settings that you specify for the connection. All adapter services use connection pooling.

A connection pool is a collection of connections with the same set of attributes. Integration Server maintains connection pools in memory. Connection pools improve performance by enabling adapter services to re-use the open connections instead of opening new connections.

Run-Time Behavior of Connection Pools

When you enable a connection, Integration Server initializes the connection pool, creating the number of connection instances you specified in the connection's **Minimum Pool Size** field when you configured the connection. Whenever an adapter service needs a connection, Integration Server provides a connection from the pool. If no connections are available in the pool, and the maximum pool size has not been reached, the server creates one or more new connections (according to the number specified in the **Pool Increment Size** field) and adds them to the connection pool. If the pool is full (as specified in **Minimum Pool Size** field), the requesting service will wait for Integration Server to obtain a connection, up to the length of time specified in the **Block Timeout** field, until a connection becomes available. Periodically, Integration Server inspects the pool and removes inactive connections that have exceeded the expiration period that you specified in the **Expire Timeout** field.

If initialization of the connection pool fails because of a network connection failure or some other type of exception, you can enable the system to retry the initialization any number of times, at specified intervals. For information about configuring connections, see [“Configuring an Adapter for Apache HBase Connection” on page 32](#).

Built-In Services for Connections

Integration Server provides built-in services that enable you to programmatically control connections. You can use them to enable and disable a connection, and to return usage statistics and the current state (Enabled or Disabled) and error status for a connection. These services are located in the WmART package, in the `pub.art.connection` folder.

The `setAdapterServiceNodeConnection` built-in service enables you to change the connection associated with an adapter service respectively. For more information, see [“Change the Connection Associated with an Adapter Service at Design Time” on page 15](#).

For details, see the *webMethods Integration Server Built-In Services Reference* for your release.

Adapter Services

To use Adapter for Apache HBase, you create adapter services. Adapter services allow you to connect to the adapter's resource and initiate an operation on the resource from Integration Server

You call adapter services from flow or Java services to interact with HBase. The adapter services perform database operations by calling Apache HBase Java API. Integration Server then uses adapter connections that you defined earlier to execute the adapter services. For details, see [“Adapter Service Processing” on page 15](#).

Adapter services are based on templates provided with Adapter for Apache HBase. Each template represents a specific technique for doing work on a resource, such as using the Retrieve service template to retrieve specified information from the database

An adapter service template contains all the code necessary for interacting with the resource but without the data specifications. You provide these specifications when you create a new adapter service.

Creating a new service from an adapter service template is straightforward. Using Software AG Designer, you assign the service a default adapter connection.

After you select the connection for the adapter service, you select the adapter service template and supply the data specifications using Designer. Some familiarity with using Designer is required. For more information, see the *webMethods Service Development Help* for your release.

Adapter for Apache HBase provides the following adapter service templates:

Adapter Service Template	Description
Retrieve Operation Service	Retrieves records from HBase table and returns the values of column qualifiers of those records. For instructions about configuring the service, see “Configuring Retrieve Operation Service” on page 38 .

Adapter Service Template	Description
Delete Operation service	<p>Deletes a specific row, column family, or column qualifier completely from a selected table.</p> <p>For instructions about configuring the service, see “Configuring Delete Operation Service” on page 44.</p>
Upsert Operation service	<p>Either inserts the records into or updates the records of an HBase table. If the record does not already exist, it is inserted. If the record exists, it is updated.</p> <p>For instructions about configuring the service, see “Configuring Upsert Operation Service” on page 46.</p>
Retrieve Operation With Comparator service	<p>Retrieves records from HBase table and returns the values of column qualifiers of those records based on the provided filters, standard set of compound operators, and comparators form hierarchical queries.</p> <p>For instructions about configuring the service, see “Configuring Retrieve Operation With Comparator Service” on page 49.</p>

Using Adapter Services

The following table lists the tasks required to use the adapter services.

1. Create an adapter connection using Integration Server Administrator. For details, see [“Overview of Adapter for Apache HBase Connections” on page 32.](#)
2. Select the appropriate adapter service template and configure the adapter service using Designer. Depending on the type of adapter service, you specify:
 - The adapter connection
 - The input fields and types as needed
 - The output fields and types as needed

For more information about configuring adapter services, see [“Overview of Adapter Services” on page 38.](#)
3. If you plan to use an Integration Server flow or Java service to invoke the adapter service, design the flow or Java service to use this adapter service using Designer.
4. Manage the adapter service using Integration Server and Designer.

Change the Connection Associated with an Adapter Service at Design Time

Integration Server provides built-in services that you can use at design time to change the connection associated with an adapter service. The built-in services and `setAdapterServiceNodeConnection` are provided in the WmART package's `pub.art.service` folder. Using this function, you can change the specific connection associated with an adapter service at design time so that you need not create and maintain multiple adapter services.

Note:

The `setAdapterServiceNodeConnection` can run at design time only. Do not use it within an Integration Server flow or Java service. You must run the services directly from Designer by selecting a service and running it.

For details, see the *webMethods Integration Server Built-In Services Reference* for your release.

Other built-in services enable you to control connections. For more information, see [“Built-In Services for Connections” on page 13](#).

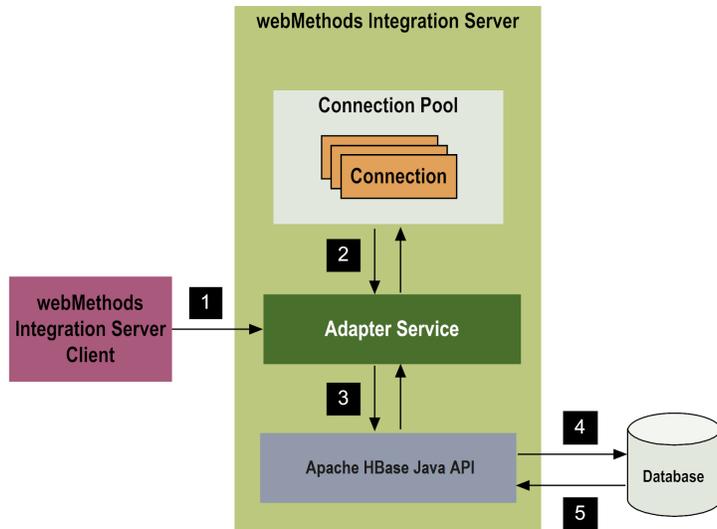
Change the Connection Associated with an Adapter Service at Run Time

Integration Server enables you to dynamically select the connection that a service uses to interact with the adapter's resource. This feature enables one service to interact with multiple, and similar backend resources.

For example, a service can be defined to use a default connection that interacts with your company's production database. However, at run time you can override the default connection and instead use another connection to interact with the company's test database.

Adapter Service Processing

The following diagram illustrates how Adapter for Apache HBase processes adapter services at run time.



Step	Description
1	An Integration Server client, typically using a flow or Java service, invokes a Adapter for Apache HBase service on Integration Server to perform an operation on a HBase. You configured the adapter service earlier using Designer.
2	The adapter service gets a connection from the service's connection pool. Adapter connections contain connection information of the database.
3	The adapter service uses the Apache HBase Java API to connect to the database. You created and enabled the adapter connection earlier using Integration Server Administrator.
4	All adapter services performs the CRUD operation against Apache HBase. <ul style="list-style-type: none"> ■ For Retrieve, Delete, and Upsert services, the adapter service executes a query statement against the database.
5	Depending on the adapter service type, such as a Retrieve service, the adapter service may return data to Integration Server. <ul style="list-style-type: none"> ■ If the operation is successful, the service returns the output from the Apache HBase Java API, if applicable. ■ If the operation is unsuccessful, the service returns an error such as an AdapterException. If the database throws an exception while performing the adapter service's operation, the adapter passes the exception to the Integration Server logs. <p>For more information about how the adapter handles exceptions, see “Overview of Logging and Exception Handling” on page 64.</p>

Viewing the Adapter's Update Level

You can view the list of updates that have been applied to the adapter. The list of updates appears in the **Updates** field on the adapter's About page in Integration Server Administrator.

Controlling Pagination

When using the adapter on Integration Server 9.8 and later, you can control the number of items that are displayed on the adapter Connections screen. By default, 10 items are displayed per page. Click **Next** and **Previous** to move through the pages, or click a page number to go directly to a page.

To change the number of items displayed per page, set the `watt.art.page.size` property and specify a different number of items.

➤ To set the number of items per page

1. From Integration Server Administrator, click **Settings>Extended** .
2. Click **Edit Extended Settings**. In the Extended Settings editor, add or update the `watt.art.page.size` property to specify the preferred number of items to display per page. For example, to display 50 items per page, specify:

```
watt.art.page.size=50
```

3. Click **Save Changes**. The property appears in the Extended Settings list.

For more information about working with extended configuration settings, see the *webMethods Integration Server Administrator's Guide* for your release.

2 Installing and Uninstalling Adapter for Apache HBase

- Overview of installing and uninstalling Adapter for Apache HBase 20
- Requirements 20
- The Integration Server Home Directory 20
- Installing Adapter for Apache HBase 20
- Uninstalling Adapter for Apache HBase 21
- Migrating Retrieve Operation to Retrieve Operation With Comparator Service 22

Overview of installing and uninstalling Adapter for Apache HBase

This chapter explains how to install, and uninstall webMethods Adapter 9.8 for Apache HBase. The instructions use the Software AG Installer and Software AG Uninstaller the wizards. For complete information about the wizards or other installation methods, or to install other webMethods products, see the *Installing webMethods Products On Premises* for your release.

Requirements

For a list of operating systems, and webMethods products supported by Adapter for Apache HBase, see *webMethods Adapters System Requirements* .

Adapter for Apache HBase has no hardware requirements beyond those of its host Integration Server.

The Integration Server Home Directory

Beginning with Integration Server 9.8, you can create and run multiple Integration Server instances under a single installation directory. Each Integration Server instance has a home directory under *Integration Server_directory \instances\instance_name* that contains the packages, configuration files, log files, and updates for the instance.

For more information about running multiple Integration Server instances, see the *webMethods Integration Server Administrator's Guide* for your release.

This guide uses the *packages_directory* as the home directory in Integration Server classpaths. For Integration Server 9.8 and above, the *packages_directory* is *Integration Server_directory \instances\instance_name\packages* directory.

Installing Adapter for Apache HBase

Note:

If you are installing Adapter for Apache HBase in a clustered environment, you must install the adapter on each Integration Server in the cluster, and each installation must be identical. For more information about working with Adapter for Apache HBase in a clustered environment, see [“Adapter for Apache HBase in a Clustered Environment” on page 27](#).

➤ To install Adapter for Apache HBase

1. Download Installer from the [Empower Product Support website](#).
2. If you are installing the adapter on an existing Integration Server, shut down the Integration Server.
3. Start the Installer wizard.

4. Choose the webMethods release that includes the Integration Server on which you want to install the adapter. For example, if you want to install the adapter on Integration Server 9.8, choose the 9.8 release.
5. Specify the installation directory as follows:
 - If you are installing on an existing Integration Server, specify the webMethods installation directory that contains the host Integration Server.
 - If you are installing on an existing Integration Server and the adapter, specify the installation directory to use.
6. In the product selection list, select **Adapters > webMethods Adapter 9.8 for Apache HBase**.
 If you are Integration Server using 9.8 and above, you can choose to install the package in the default instance. In this case, Software AG Installer installs the adapter in both locations, *Integration Server_directory* \packages and the default instance packages directory located in *Integration Server_directory* \instances\default\packages.
7. To download the documentation for the adapter, go to [Software AG Documentation website](#).
8. HBase jars files has to be copied to WmHbaseAdapter/code/jars folder, in the following ways:
 - Copy all the jar files from Apache HBase's *lib* folder.

Note:
Do not copy the following jars, while connecting to Apache HBase 1.2.2:

 - xml-apis-1.3.03.jar
 - xml-apis-ext-1.3.04.jar
 - Copy all the jar files similar to Apache HBase from the distributors(Cloudera CDH), with which connections need to be established.
9. After the installation completes, close the Installer and start the host Integration Server.

Uninstalling Adapter for Apache HBase

> To uninstall Adapter for Apache HBase

1. Shut down the host Integration Server. You do not need to shut down any other webMethods products or applications that are running on your machine.
2. Start Software AG Uninstaller, selecting the webMethods installation directory that contains the host Integration Server.
3. In the product selection list, select **Adapters>webMethods Adapter 9.8 for Apache HBase**. You can also choose to uninstall documentation.

4. After Uninstaller completes, restart the host Integration Server.

Uninstaller removes all Adapter for Apache HBase-related files that were installed. However, Uninstaller does not delete files created after you installed the adapter (for example, user-created or configuration files), nor does it delete the adapter directory structure. You can go to the *Integration Server_directory* \packages directory and *Integration Server_directory* \instances\default\packages directory. Delete the WmHBaseAdapter directory.

Migrating Retrieve Operation to Retrieve Operation With Comparator Service

In Adapter for Apache HBase, the new Retrieve Operation With Comparator service is introduced with more capabilities to form retrieve queries. Hence the existing Retrieve Operation adapter service will be deprecated for the future releases.

The migration utility service is provided to move the existing Retrieve Operation adapter service to new Retrieve Operation With Comparator service template.

Use the `wm.wmHBaseAdapter.admin:migrateServices` migration utility service to migrate the adapter service from Retrieve Operation to Retrieve Operation With Comparator Service.

3 Package Management

- Overview of Package Management 24
- Adapter for Apache HBase Package Management 24
- Group Access Control 27
- Adapter for Apache HBase in a Clustered Environment 27

Overview of Package Management

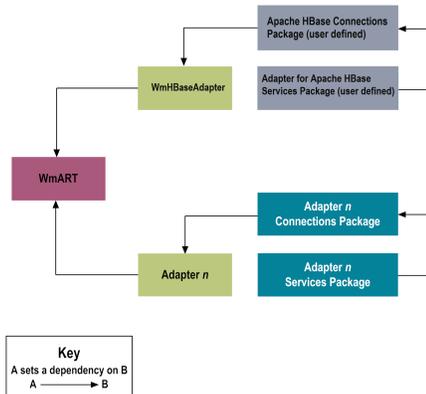
The following sections describe how to set up and manage your webMethods Adapter for Apache HBase packages and to set up Access Control Lists (ACLs).

Adapter for Apache HBase Package Management

Adapter for Apache HBase is provided as a package called WmHBaseAdapter. You can manage the WmHBaseAdapter package as you would manage any package on webMethods Integration Server.

When you create connections, and adapter services, define them in user-defined packages rather than in the WmHBaseAdapter package. Doing so will allow you to manage the package more easily.

As you create user-defined packages in which to store connections, and adapter services, use the package management functionality provided in Software AG Designer and set the user-defined packages to have a dependency on the WmHBaseAdapter package. That way, when the WmHBaseAdapter package loads or reloads, the user-defined packages load automatically. See the following diagram:



Package management tasks include:

- [Setting package dependencies \(see “Package Dependency Requirements and Guidelines” on page 24\)](#)
- [“Enabling Packages” on page 25](#)
- [“Importing and Exporting Packages” on page 26](#)
- [“Group Access Control” on page 27](#)

Package Dependency Requirements and Guidelines

This section contains a list of dependency requirements and guidelines for user-defined packages. For instructions for setting package dependencies, see the *webMethods Service Development Help* for your release.

- A user-defined package must have a dependency on its associated adapter package, WmHBaseAdapter. (The WmHBaseAdapter package has a dependency on the WmART package.)
- Package dependencies ensure that at start-up the Integration Server automatically loads or reloads all packages respectively: the WmART package, the adapter package, and the user-defined packages. The WmART package is automatically installed when you install Integration Server. You should not need to manually reload the WmART package.
- If the connections and adapter services of an adapter are defined in different packages, then:
 - A package that contains the connections must have a dependency on the adapter package.
 - Packages that contain adapter services must have a dependency on their associated connection package.
- Keep connections for different adapters in separate packages so that you do not create interdependencies between adapters. If a package contains connections for two different adapters, and you reload one of the adapter packages, the connections for both adapters will reload automatically.
- Integration Server will not allow you to enable a package if it has a dependency on another package which is disabled. That is, before you can enable your package, you must enable all packages on which your package depends. For information about enabling packages, see [“Enabling Packages” on page 25](#).
- Integration Server will allow you to disable a package even if another package that is enabled has a dependency on it. Therefore, you must manually disable any user-defined packages that have a dependency on the adapter package before you disable the adapter package. For information about disabling packages, see [“Disabling Packages” on page 26](#).
- You can provide same name for the connections, and adapter services, if they are in different folders and packages.

Enabling Packages

All packages are automatically enabled by default. Use the following procedure when you want to enable a package that was previously disabled.

➤ To enable a package

1. Open Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Click **No** in the **Enabled** column. The server displays **Yes** in the **Enabled** column.

Note:

Enabling an adapter package will not cause its associated user-defined packages to be reloaded.

Important:

Before you manually enable a user-defined package, you must first enable its associated adapter package (WmHBaseAdapter).

Disabling Packages

When you want to temporarily prohibit access to the elements in a package, disable the package. When you disable a package, the server unloads all of its elements from memory. Disabling a package prevents Integration Server from loading that package at startup.

Important:

If your adapter has multiple user-defined packages, and you want to disable some of them, disable the adapter package (WmHBaseAdapter) first. Otherwise, errors will be issued when you try to access the remaining enabled user-defined packages.

> To disable a package

1. Open Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. Click **Yes** in the **Enabled** column for the package that you want to disable. The server issues a prompt to verify that you want to disable the package. Click **OK** to disable the package. When the package is disabled, the server displays **No** in the **Enabled** column.

A disabled adapter will:

- Remain disabled until you explicitly enable it using Integration Server Administrator.
- Not be listed in **Designer**.

Importing and Exporting Packages

You can import and export packages using Designer. Exporting allows you to export the package to a .zip file and save it to your hard drive. The .zip file can then be imported for use by another package.

Important:

Do not rename packages that you export; the rename function is comparable to moving a package, and when you import the renamed package, you can lose any triggers, and connections, associated with this package.

For details about importing and exporting packages, see the *webMethods Service Development Help* for your release.

Group Access Control

To control which groups have access to which adapter services, use access control lists (ACLs). For example, you can use ACLs to prevent one development group from inadvertently updating the work of another group, or to allow or deny access to services that are restricted to one group but not to others.

For information about assigning and managing ACLs, see the *webMethods Service Development Help* for your release.

Adapter for Apache HBase in a Clustered Environment

Clustering is an advanced feature of the webMethods product suite that substantially extends the reliability, availability, and scalability of Integration Server. Clustering accomplishes this by providing the infrastructure and tools to deploy multiple Integration Server as if they were a single virtual server and to deliver applications that leverage that architecture. Because this activity is transparent to the client, clustering makes multiple servers look and behave as one.

Integration Server 9.0 and higher supports the caching and clustering functionality provided by Terracotta. Caching and clustering are configured at the Integration Server level and Adapter for Apache HBase uses the caching mechanism that is enabled on Integration Server. Adapter for Apache HBase does not explicitly implement any clustering or caching beyond what is already provided by Integration Server.

With clustering, you get the following benefits:

- **Load balancing.** This feature, provided automatically when you set up a clustered environment, allows you to spread the workload over several servers, thus improving performance and scalability.
- **Failover support.** Clustering enables you to avoid a single point of failure. If a server cannot handle a request, or becomes unavailable, the request is automatically redirected to another server in the cluster.

Note:

Integration Server clustering redirects HTTP and HTTPS requests, but does not redirect FTP or SMTP requests.

- **Scalability.** You can increase your capacity even further by adding new machines running Integration Server to the cluster.

For details on Integration Server clustering, see the *webMethods Integration Server Clustering Guide* for your release.

Adapter Service Support in Clusters

Adapter services are supported in a clustered environment. In order for a cluster to handle requests identically, you should be sure that the identical service is in each server in the cluster, so that if a given service is not available, the request can be redirected and handled by another server in the cluster.

For more details about adapter services in clusters, see [“Clustering Considerations and Requirements” on page 28](#).

Replicating Packages to Integration Server

Every Integration Server in the cluster should contain an identical set of packages that you define using Adapter for Apache HBase ; that is, you should replicate the Adapter for Apache HBase services, and the connections they use.

To ensure consistency, we recommend that you create all packages on one server, and replicate them to the other servers. If you allow different servers to contain different services, you might not derive the full benefits of clustering. For example, if a client requests a service that resides in only one server, and that server is unavailable, the request cannot be successfully redirected to another server.

For information about replicating packages, see the *webMethods Integration Server Administrator’s Guide* for your release.

Clustering Considerations and Requirements

Note:

The following sections assume that you have already configured the Integration Server cluster. For details about webMethods clustering, see the *webMethods Integration Server Clustering Guide* for your release and *webMethods Integration Server Administrator’s Guide*, if they are not already running.

The following considerations and requirements apply to Adapter for Apache HBase in a clustered environment.

Requirements for Each Integration Server in a Cluster

The following table describes the requirements of each Integration Server in a given cluster:

All Integration Servers in a given cluster must have identical...	For Example
Integration Server versions	All Integration Server's in the cluster must be the same version, with the same service packs and fixes applied.
Adapter packages	All adapter packages on one Integration Server should be replicated to all other Integration Servers in the cluster.
Adapter connections	If you configure a connection to the database, this connection must appear on all servers in the cluster so that any Integration Server in the cluster can handle a given request identically.

All Integration Servers in a given cluster must have identical...

For Example
If you plan to use connection pools in a clustered environment, see [“Considerations When Configuring Connections with Connection Pooling Enabled” on page 29](#)

Adapter services If you configure a specific Retrieve Adapter Service, this same adapter service must appear on all servers in the cluster so that any Integration Server in the cluster can handle the request identically.

If you allow different Integration Servers to contain different services, you might not derive the full benefits of clustering. For example, if a client requests a service that resides on only one server, and that server is unavailable, the request cannot be successfully redirected to another server.

For information about replicating adapter packages, connections, adapter services across multiple Integration Server in a cluster, see [“Replicating Packages to Integration Server” on page 28](#)

Considerations When Installing Adapter for Apache HBase Packages

For each Integration Server in the cluster, use the standard Adapter for Apache HBase installation procedures for each machine, as described in [“Overview of installing and uninstalling Adapter for Apache HBase” on page 20](#)

Considerations When Configuring Connections with Connection Pooling Enabled

When you configure a connection that uses connection pools in a clustered environment, be sure that you do not exceed the total number of connections that can be opened simultaneously for that database.

For example, if you have a cluster of two Integration Servers with a connection configured to a database that supports a maximum of 100 connections opened simultaneously, the total number of connections possible at one time must not exceed 100. This means that you cannot configure a connection with an initial pool size of 100 and replicate the connection to both servers, because there could be possibly a total of 200 connections opened simultaneously to this database.

In another example, consider a connection configured with an initial pool size of 10 and a maximum pool size of 100. If you replicate this connection across a cluster with two Integration Servers, it is possible for the connection pool size on both servers to exceed the maximum number of database connections that can be open at one time.

For information about configuring connections for Adapter for Apache HBase, see [“Overview of Adapter for Apache HBase Connections” on page 32](#).

For more general information about connection pools, see the *webMethods Integration Server Administrator's Guide* for your release.

4 Adapter for Apache HBase Connections

- Overview of Adapter for Apache HBase Connections 32
- Before Configuring or Managing Adapter Connection 32
- Configuring an Adapter for Apache HBase Connection 32
- Configuring an Adapter for Apache HBase Kerberos Connection 35

Overview of Adapter for Apache HBase Connections

This chapter describes how to configure and manage Adapter for Apache HBase connections. For more information about how adapter connections work, see [“Adapter Connections” on page 12](#).

Before Configuring or Managing Adapter Connection

Perform the following steps before configuring or managing adapter connections.

➤ To prepare to configure or manage adapter connections

1. Install webMethods Integration Server and Adapter for Apache HBase on the same machine. For details, see [“Overview of installing and uninstalling Adapter for Apache HBase ” on page 20](#).
2. Make sure you have Integration Server administrator privileges so that you can access Adapter for Apache HBase's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.
3. Start your Integration Server and Integration Server Administrator, if they are not already running.
4. Using Integration Server Administrator, make sure the WmHBaseAdapter package is enabled. For instructions, see [“Enabling Packages” on page 25](#).
5. Using Designer, create a user-defined package to contain the connection, if you have not done so. For more information about managing packages for the adapter, see [“Overview of Package Management” on page 24](#).

Configuring an Adapter for Apache HBase Connection

When you configure Adapter for Apache HBase , you specify information that Integration Server uses to connect to an Apache HBase system. You can configure Adapter for Apache HBase connections manually using the Integration Server Administrator screen.

➤ To configure an adapter connection

1. In the **Adapters** menu of Integration Server Administrator's navigation area, click **webMethods Adapter for Apache HBase**.
2. On the Connections screen, click **Configure New Connection**.
3. On the Connection Types screen, click **webMethods Adapter for Apache HBase Connection** to display the Configure Connection Type screen.

4. In the **webMethods Adapter for Apache HBase** section, use the following fields:

Field	Description/Action
Package	The package in which to create the connection. You must create the package using Designer before you can specify the package using this parameter. For general information about creating packages, see the <i>webMethods Service Development Help</i> for your release. Note: Configure the connection in a user-defined package rather than in the adapter's package. For other important considerations when creating packages for Adapter for Apache HBase, see “Adapter for Apache HBase Package Management” on page 24
Folder Name	Specifies the folder in which you create the connection.
Connection Alias Name	Specifies the name you want to give to the connection. Connection names cannot have spaces or use special characters reserved by Integration Server and Designer. For more information about the use of special characters in package, folder, and element names, see the <i>webMethods Service Development Help</i> for your release.
Zookeeper Host Name	Specifies the name of the server that hosts the Zookeeper.
Zookeeper Port Number	Specify the port number that the connection uses to connect to the Zookeeper.
Other Properties	Specifies the HBase client related properties for additional configuration. Use the following format: <code>propertyName1=value;propertyName2=value</code>

You can set the Apache HBase cluster in one of the following ways:

- **hbase-site.xml:** You can set the cluster related configurations in `hbase-site.xml` file and place it in *Integration Server_directory* \instances\instance_name\packages\WmHBaseAdapter\config directory.

Note:

The configurations mentioned in `hbase-site.xml` file are applied to all the connections of Adapter for Apache HBase.

- **Other Properties:** If you need to set any cluster related configurations for a specific connection, specify it in the **Other Properties** field of the Configure Connection Type screen.

Note:

Configurations specified in the **Other Properties** field takes precedence over the `hbase-site.xml` file.

5. In the **Connection Management Properties** section, use the following fields:

Field	Description/Action
Enable Connection Pooling	<p>Enables the connection to use connection pooling. For more information about connection pooling, see “Adapter Connections” on page 12.</p> <p>Note: If you plan to enable connection pooling in a clustered environment, consider the connection pool size.</p>
Minimum Pool Size	If connection pooling is enabled, this field specifies the number of connections to create when the connection is enabled. The adapter will keep open the number of connections you configure here regardless of whether these connections become idle.
Maximum Pool Size	If connection pooling is enabled, this field specifies the maximum number of connections that can exist at one time in the connection pool.
Pool Increment Size	If connection pooling is enabled, this field specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size.
Block Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that Integration Server will wait to obtain a connection with the database before it times out and returns an error. For example, you have a pool with Maximum Pool Size of 20. If you receive 30 simultaneous requests for a connection, 10 requests will be waiting for a connection from the pool. If you set the Block Timeout to 5000, the 10 requests will wait for a connection for 5 seconds before they time out and return an error. If the services using the connections require 10 seconds to complete and return connections to the pool, the pending requests will fail and return an error message stating that no connections are available. If you set the Block Timeout value too high, you may encounter problems during error conditions. If a request contains errors that delay the response, other requests will not be sent. This setting should be tuned in conjunction with the Maximum Pool Size to accommodate such bursts in processing.
Expire Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. The connection pool will remove inactive connections until the number of connections in the pool is equal to the Minimum Pool Size . The inactivity timer for a connection is reset when the connection is used by the adapter.

Field	Description/Action
	<p>If you set the Expire Timeout value too high, you may have a number of unused inactive connections in the pool. This consumes local memory and a connection on your backend resource. This could have an adverse effect if your resource has a limited number of connections.</p> <p>If you set the Expire Timeout value too low, performance could degrade because of the increased activity of creating and closing connections. This setting should be tuned in conjunction with the Minimum Pool Size to avoid excessive opening/closing of connections during normal processing.</p>
Startup Retry Count	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. The default is 0.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

6. Click **Save Connection**.

The connection you created appears on the adapter's Connections screen and in Designer.

You can enable a connection only if the parameters for the connection are valid.

Configuring an Adapter for Apache HBase Kerberos Connection

➤ To configure an Adapter for Apache HBase Kerberos connection

1. In the **Adapters** menu of Integration Server Administrator's navigation area, click **webMethods Adapter for Apache HBase**.
2. On the Connections screen, click **Configure New Connection**.
3. On the Connection Types screen, click **webMethods Adapter for Apache HBase Kerberos Connection** to display the Configure Connection Type screen.
4. In the **webMethods Adapter for Apache HBase**, use the following fields:

Field	Description/Action
Package	The package in which to create the connection. You must create the package using Designer before you can specify the package using this parameter. For general information about creating packages, see the <i>webMethods Service Development Help</i> for your release.

Field	Description/Action
	<p>Note: Configure the connection in a user-defined package rather than in the adapter's package. For other important considerations when creating packages for Adapter for Apache HBase, see “Adapter for Apache HBase Package Management” on page 24</p>
Folder Name	Specifies the folder to which you create the connection.
Connection Alias Name	Specifies the name you want to give to the connection. Connection names cannot have spaces or use special characters reserved by Integration Server and Designer. For more information about the use of special characters in package, folder, and element names, see the <i>webMethods Service Development Help</i> for your release.
Zookeeper Host Name	Specifies the name of the server that hosts the Zookeeper.
Zookeeper Port Number	Specify the port number that the connection uses to connect to the Zookeeper.
Principal	<p>The name of the principal to use for Kerberos authentication.</p> <p>Specify the fully qualified domain name of the principal in the <i>hbase/fully.qualified.domain.name@REALM</i> format.</p>
Keytab File Path	<p>The local path to the keytab file that contains information about the specified principal.</p> <p>For example, you can specify a keytab file at the following path: path: C:/users/foo/hbase.keytab</p>
Other Properties	<p>Specifies the HBase client related properties for additional configuration. Use the following format:</p> <p>propertyName1=value;propertyName2=value</p>

5 Adapter Services

■ Overview of Adapter Services	38
■ Before Configuring or Managing Adapter Services	38
■ Configuring Retrieve Operation Service	38
■ Configuring Delete Operation Service	44
■ Configuring Upsert Operation Service	46
■ Configuring Retrieve Operation With Comparator Service	49

Overview of Adapter Services

This chapter describes how to configure and manage Adapter for Apache HBase services. For detailed descriptions of the available Adapter for Apache HBase services, see [“Adapter Services” on page 13](#).

Before Configuring or Managing Adapter Services

Perform the following steps before configuring or managing adapter services.

➤ To prepare to configure or manage Adapter for Apache HBase services

1. Start the Integration Server and Integration Server Administrator, if they are not already running.
2. Make sure you have administrator privileges so that you can access the Adapter for Apache HBase administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator’s Guide* for your release.
3. Using Integration Server Administrator, make sure the WmHBaseAdapter package is enabled. For instructions, see [“Enabling Packages” on page 25](#).
4. Using Integration Server Administrator, configure an adapter connection to use with the adapter service. For instructions, see [“Overview of Adapter for Apache HBase Connections” on page 32](#).

Note:

Integration Server provides a built-in services you can use at design time to change the connection associated with an adapter service. For more information, see [“Change the Connection Associated with an Adapter Service at Design Time” on page 15](#).

5. Start Designer if it is not already running.
6. Using Designer, create a user-defined package to contain the services, if you have not already done so. When you configure adapter services, you should always define them in user-defined packages rather than in the WmHBaseAdapter package. For more information about managing packages for the adapter, see [“Overview of Package Management” on page 24](#).

Configuring Retrieve Operation Service

Use the **Retrieve Operation** template to configure an adapter service that retrieves selected fields from the records of a HBase table. This service can either retrieve all the records from the selected table or retrieve any required records based on the filters provided. For more information about adapter services, see [“Using Adapter Services” on page 14](#).

Warning:

This service may get deprecated in future. Any feature or enhancement request for this service will not be considered. Usage of Retrieve service with comparator is recommended.

Be sure to review the section [“Before Configuring or Managing Adapter Services”](#) on page 38, before you configure adapter services.

➤ **To configure an adapter service using the Retrieve Operation template**

1. In Designer, right-click the package in which the service should be contained and select **New > Adapter Service**.
2. Select the parent namespace, type a name for the adapter service, and click **Next**.
3. Select **Adapter for Apache HBase** as the adapter type and click **Next**.
4. Select the appropriate **Adapter Connection Name** and click **Next**.
5. From the list of available templates, select the **Retrieve Operation** template and click **Finish**.

The adapter service editor for the selected adapter service appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template** as necessary.

6. Select the **Schema** tab to configure the database table, using the following fields:

Field	Field Description/Action
HBase Table	Select a table name from the list of tables.
Column Family	Displays the list of column families for the selected table name.
Select	Select the check box for the corresponding column family to query.
Maximum Version	Specify the maximum number of versioned values for a corresponding column family.

Sampling Range Specify the number of records to be scanned to get a minimum set of column qualifiers. These column qualifiers are listed in the **Columns** tab.

Value can be entered in the following formats:

- Input value can be an integer and it should be greater than 0.

For example, if you provide Sampling Range: 10, then the service will scan initial 10 records of the selected table.

- Input value can be the range between two row ids separated by hyphen (-). The start row id and the end row id should be valid row ids, present in the selected table. The start Row Id should be lexicographically smaller than the end Row Id.

Field	Field Description/Action
	<p>For example, if you provide Sampling Range: row100-row1000, then the service will scan all the records which contain row ids between row100 to row1000</p> <p>Note: Sampling is not performed for the values which do not contain the above formats, and the list of column qualifiers under Columns tab will remain empty.</p>
Versions	<p>Specify the number of versions for the selected table, to retrieve more than one value for a particular column qualifier. The default value is 1. Use the default value to get the latest value of the column qualifiers.</p> <p>Note: Minimum value of version should be 1, to get a valid output.</p>
Output Structure	<p>Select either of the following Output Structure:</p> <ul style="list-style-type: none"> ■ By Column Family ■ By Row Id <p>Note: If you do not make an entry for a column qualifier in Columns tab, then the Output Structure of the table remains the same, and returns the complete list of Row Ids of the selected table.</p>

7. Use the **Columns** tab to define the column qualifiers using the following fields:
- a. Use the  icon to create a new row as needed. You can use the  icon to fill in all column families to the table.

Use the following fields:

Field	Description/Action
Column Family	Provides the list of column families based on the selection made in the Schema tab.
Select Column Qualifier	<p>Select the column qualifier which can be retrieved for the corresponding column family.</p> <p>The Select Column Qualifier field is displayed as a drop down list. These values are retrieved based on the Sampling Range provided in the Schema tab.</p>
Column Qualifier Type	The Select Column Qualifier field supports the following data types:

Field	Description/Action
	<ul style="list-style-type: none"> ■ java.lang.String ■ byte array ■ java.lang.Object
Enter Column Qualifier	Specify the column qualifiers which are not listed in the Select Column Qualifier field.

Note:

If you change the values of **Table Name** and **Column Family** of **Schema** tab then, the following fields for **Columns** tab changes:

- The values of **Column Family** and **Select Column Qualifier** changes automatically.
- The **Column Family** changes automatically, but you need to manually change the values of **Enter Column Qualifier**.

Note:

You must select an appropriate data type for the Column Qualifiers, based on the data type being inserted.

8. Use the **WHERE** tab to specify the conditions for retrieving records:
 - a. Select the  icon to define new WHERE clause fields.
 - b. Select a logical operator from the **AND/OR** field, an Operator as needed, and specify values for the following fields:

Name	Description/Action
AND/OR	Input Field.
Parameter Key	<p>Defines a unique key.</p> <p>Rows in both tables in the WHERE clause will be inter dependent if it contains same parameter key.</p> <p>For example, if you delete a row having parameter key as 1, then it automatically deletes the row having the same parameter key from the other table.</p> <p>Note: By default, the number of unique keys are between 1 to 25. To increase the number of unique keys, set the watt property as, <code>watt.hbase.filters.limit=30</code>.</p>
Filters	Displays the list of different filters. Different types of filters are listed below:

Name	Description/Action
	<ul style="list-style-type: none">■ Row Id- To retrieve records based on Row Id.■ Row Id Prefix- To retrieve records based on Prefix of the Row Id. <p>Note: Row Id Prefix works only with EQUAL operator.</p> <ul style="list-style-type: none">■ Timestamp- To retrieve records based on Timestamp, where Timestamp is the time when the record is inserted. You can enter a comma separated multiple Timestamps. Do not enter white space characters after a comma. <p>Note: Timestamp filter works only with EQUAL operator.</p> <ul style="list-style-type: none">■ Value- To retrieve all the records, which consist of entered value of any column qualifier. <p>Note: If you select the number of versions to be retrieved as 1, then the latest version for the row matching the value filter is retrieved even if the data passed in Value filter appears in any other version in the same row.</p> <ul style="list-style-type: none">■ Column Qualifier based filters - Lists all the column qualifiers from the Columns tab. <p>Note: The Value and Column Qualifier based filters supports the following data types:</p> <ul style="list-style-type: none">■ java.lang.String■ byte array■ java.lang.Object

Operator The following are the operators used:

- =
- <
- >
- <=
- >=
- IN
- CONTAINS

Name	Description/Action
	■ CONTAINS KEY
Input Field Type	The data type of the Input Field . You can change this type if needed.
Input Field	The default value is ?, which acts as a placeholder for the variable so that you can set the input variable for that column at run time, or get input external to this adapter service. You can also type a fixed value in this field now or at run time. If you type a fixed value, be sure that it is valid, or an exception will be generated at run time.

The adapter automatically generates values for the following fields:

Field	Description/Action
Parameter Key	The unique key for those rows in the above table which takes input at runtime.
Filters (second occurrence of this field)	The name of the column you need to use in the WHERE clause.
Input Field type	The corresponding data type of an Input Field .
Input Field (second occurrence of this field)	The name of the input field. By default the name combines the values of Parameter and Field fields. However, you can also choose to specify any custom value.

- c. If necessary, use the or icons to change the order of the WHERE clause to ensure the parameters are parsed in the correct order.
- d. Repeat this procedure until you have specified all WHERE parameters.

Note:

If u do not pass any input values or if you explicitly pass the value as a NULL character, then the corresponding filters are skipped.

Note:

You must select an appropriate data type for the Column Qualifier filters and Value filters, based on the data type being inserted.

9. To verify input or output information for the service, use the **Input/Output** tab as required.
10. From the **File** menu, select **Save**.

Configuring Delete Operation Service

Use the **Delete Operation** template to configure an adapter service that deletes a specific row, column family, or column qualifier completely from a selected table. For more information about adapter services, see [“Using Adapter Services” on page 14](#).

Be sure to review the section [“Before Configuring or Managing Adapter Services” on page 38](#), before you configure adapter services.

HBase client API provides delete feature at following three levels:

- **Column Qualifier level** :- If you enter any column qualifier in the Columns tab, then the values of that column qualifier are deleted for all the input row ids provided during runtime.
- **Column Family level**: - If you do not enter a value for column qualifier in Columns tab, and if you select a column family in Schema tab, then the values of that column family are deleted for all the input row ids provided during runtime.
- **Complete Row**: - If you do not enter a column qualifier and column family in the table, then the entire records of the input row ids are deleted.

➤ To configure an adapter service using the Delete Operation template

1. In Designer, right-click the package in which the service should be contained and select **New > Adapter Service**.
2. Select the parent namespace, type a name for the adapter service, and click **Next**.
3. Select **Adapter for Apache HBase** as the adapter type and click **Next**.
4. Select the appropriate **Adapter Connection Name** and click **Next**.
5. From the list of available templates, select the **Delete Operation** template and click **Finish**

The adapter service editor for the adapter service appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template** as necessary.

6. Select the **Schema** tab to configure the database table, using the following fields:

Field	Description/Action
HBase Table	Select a table name from the list of tables.
Column Family	Displays the list of column families for the selected table name.
Select	Select the check box for the corresponding column family to query.

Field	Description/Action
Sampling Range	<p>Specify the number of records to be scanned to get a minimum set of column qualifiers. These column qualifiers are listed in the Columns tab.</p> <p>Value can be entered in the following formats:</p> <ul style="list-style-type: none"> Input value can be an integer and it should be greater than 0. <p>For example, if you provide Sampling Range: 10, then the service will scan initial 10 records of the selected table.</p> <ul style="list-style-type: none"> Input value can be the range between two row ids separated by hyphen (-). The start row id and the end row id should be valid row ids, present in the selected table. The start Row Id should be lexicographically smaller than the end Row Id. <p>For example, if you provide Sampling Range: row100-row1000, then the service will scan all the records which contain row ids between row100 to row1000.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: Sampling is not performed for the values which do not contain the above formats, and the list of column qualifier under Columns tab will remain empty.</p> </div>
Timestamp Operator	<p>Used along with the Timestamp field. You can select a Timestamp Operator, which can either be EQUAL or LESS_OR_EQUAL. If you select LESS_OR_EQUAL, then all the records whose Timestamp value is less than or equal to the entered Timestamp value gets deleted. The default is EQUAL.</p>

7. Use the **Columns** tab to define the column qualifiers using the following fields:

- Use the  icon to create a new row as needed. You can use the  icon to fill in all column families to the table.

Use the following fields:

Field	Description/Action
Row ID Data Type	Select the data type to retrieve the required row id.
Column Family	Provides the list of column families based on the selection made in the Schema tab.
Select Column Qualifier	Select the column qualifier which can be retrieved for the corresponding column family.

Field	Description/Action
	The Select Column Qualifier field is displayed as a drop down list. These values are retrieved based on the sampling range provided in the Schema tab.
Column Qualifier Type	The Select Column Qualifier field supports the following data types: <ul style="list-style-type: none"> ■ java.lang.String ■ byte array
Enter Column Qualifier	Specify the column qualifiers which are not listed in the Select Column Qualifier field.

Note:

If you change the values of **Table Name** and **Column Family** of **Schema** tab then, the following fields for **Columns** tab changes:

- The values of **Column Family** and **Select Column Qualifier** changes automatically.
- The **Column Family** changes automatically, but you need to manually change the values of **Enter Column Qualifier**.

- To verify input or output information for the service, use the **Input/Output** tab as required.

The below fields are auto generated under input section of **Input/Output** tab:

Field	Description/Action
Row Ids	Specify the list of Row Ids to delete the corresponding records. <p>Note: It is mandatory to provide Row Ids as input to delete the records.</p>
Timestamp	Enter the timestamp to delete a particular version of the records. <p>Note: It works along with Timestamp Operator.</p>

- From the **File** menu, select **Save**.

Configuring Upsert Operation Service

Use the **Upsert Operation** template to configure an adapter service that inserts or updates records of a table. If the Row Id does not exist, then the upsert operation inserts it as a new record. If the Row Id exists, then the upsert operation updates the respective records in the table. This service can be performed on single or multiple records. For more information about adapter services, see [“Using Adapter Services” on page 14](#).

Be sure to review the section [“Before Configuring or Managing Adapter Services”](#) on page 38, before you configure adapter services.

➤ **To configure an adapter service using the Upsert Operation template**

1. In Designer, right-click the package in which the service should be contained and select **New > Adapter Service**.
2. Select the parent namespace, type a name for the adapter service, and click **Next**.
3. Select **Adapter for Apache HBase** as the adapter type and click **Next**.
4. Select the appropriate **Adapter Connection Name** and click **Next**.
5. From the list of available templates, select the **Upsert Operation** template and click Finish.

The adapter service editor for the adapter service appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template** as necessary.

6. Select the **Schema** tab to configure the database table, using the following fields:

Field	Description/Action
HBase Table	Select a table name from the list of tables.
Column Family	Displays the list of column families for the selected table name.
Select	Select the check box for the corresponding column family to query.
Sampling Range	Specify the number of records to be scanned to get minimum set of column qualifiers. These column qualifiers are listed in the Columns tab.

Value can be entered in the following formats:

- Input value can be an integer and it should be greater than 0.
For example, if you provide Sampling Range: 10, then the service will scan initial 10 records of the selected table.
- Input value can be the range between two row ids separated by hyphen (-). The start row id and the end row id should be valid row ids, present in the selected table. The start Row Id should be lexicographically smaller than the end Row Id.

For example, if you provide Sampling Range: row100-row1000, then the service will scan all the records which contain row ids between row100 to row1000.

Field	Description/Action
	<p>Note: Sampling is not performed for the values which do not contain the above formats, and the list of column qualifier under Columns tab will remain empty.</p>

7. Use the **Columns** tab to define the column qualifiers using the following fields:

Use the  icon to create a new row as needed. You can use the  icon to fill in all rows to the table.

Use the following fields:

Field	Description/Action
Row ID Data Type	Select the data type to retrieve the required row id.
Column Family	Provides the list of column families based on the selection made in the Schema tab.
Select Column Qualifier	<p>Select the column qualifier which can be retrieved for the corresponding column family.</p> <p>The Select Column Qualifier field is displayed as a drop down list. These values are retrieved based on the sampling range provided in the Schema tab.</p> <p>For lower section, you can specify the column qualifiers which are not listed in the column qualifier field of upper section.</p>
Column Qualifier Type	<p>The Select Column Qualifier field supports the following data types:</p> <ul style="list-style-type: none"> ■ java.lang.String ■ byte array
Enter Column Qualifier	Specify the column qualifiers which are not listed in the Select Column Qualifier field.

Note:

If you change the values of **Table Name** and **Column Family** of **Schema** tab then, the following fields for **Columns** tab changes:

- The values of **Column Family** and **Select Column Qualifier** changes automatically.
- The **Column Family** changes automatically, but you need to manually change the values of **Enter Column Qualifier**.

8. To verify input or output information for the service, use the **Input/Output** tab as required.

The below field is auto generated under input section of **Input/Output** tab:

Field	Description/Action
Row Ids	Specify the list of row ids to delete the corresponding records.
	Note: It is mandatory to provide row ids as input to delete the records.

- From the **File** menu, select **Save**.

Configuring Retrieve Operation With Comparator Service

Use the **Retrieve Operation With Comparator** template to configure an adapter service that retrieves selected fields from the records of a HBase table. This service can either retrieve all the records from the selected table or retrieve any required records based on filters, standard set of compound operators, and comparators are provided to form hierarchical queries. For more information about adapter services, see [“Using Adapter Services” on page 14](#).

Be sure to review the section [“Before Configuring or Managing Adapter Services” on page 38](#), before you configure adapter services.

➤ To configure an adapter service using the Retrieve Operation template

- In Designer, right-click the package in which the service should be contained and select **New > Adapter Service**.
- Select the parent namespace, type a name for the adapter service, and click **Next**.
- Select **Adapter for Apache HBase** as the adapter type and click **Next**.
- Select the appropriate **Adapter Connection Name** and click **Next**.
- From the list of available templates, select the **Retrieve Operation With Comparator Service** template and click **Finish**.

The adapter service editor for the selected adapter service appears. You can select the **Adapter Settings** tab at any time to confirm adapter service properties such as the **Adapter Name**, **Adapter Connection Name**, and **Adapter Service Template** as necessary.

- Select the **Schema** tab to configure the database table, using the following fields:

Field	Field Description/Action
HBase Table	Select a table name from the list of tables.
Column Family	Displays the list of column families for the selected table name.

Field	Field Description/Action
Select	Select the check box for the corresponding column family to query.
Maximum Version	Specify the maximum number of versioned values for a corresponding column family.
Sampling Range	<p>Specify the number of records to be scanned to get a minimum set of column qualifiers. These column qualifiers are listed in the Columns tab.</p> <p>Value can be entered in the following formats:</p> <ul style="list-style-type: none"> Input value can be an integer and it should be greater than 0. For example, if you provide Sampling Range: 10, then the service will scan initial 10 records of the selected table. Input value can be the range between two row ids separated by hyphen (-). The start row id and the end row id should be valid row ids, present in the selected table. The start Row Id should be lexicographically smaller than the end Row Id. For example, if you provide Sampling Range: row100-row1000, then the service will scan all the records which contain row ids between row100 to row1000 <p>Note: Sampling is not performed for the values which do not contain the above formats, and the list of column qualifiers under Columns tab will remain empty.</p>
Versions	<p>Specify the number of versions for the selected table, to retrieve more than one value for a particular column qualifier. The default value is 1. Use the default value to get the latest value of the column qualifiers.</p> <p>Note: Minimum value of version should be 1, to get a valid output.</p>
Output Structure	<p>Select either of the following Output Structure:</p> <ul style="list-style-type: none"> By Column Family By Row Id <p>Note: If you do not make an entry for a column qualifier in Columns tab, then the Output Structure of the table remains the same, and returns the complete list of Row Ids of the selected table.</p>

7. Use the **Columns** tab to define the column qualifiers using the following fields:

Use the  icon to create a new row as needed. You can use the  icon to fill in all column families to the table.

Use the following fields:

Field	Description/Action
Row Id Data Type	Select the data type to retrieve the required row id.
Column Family	Provides the list of column families based on the selection made in the Schema tab.
Select Column Qualifier	Select the column qualifier which can be retrieved for the corresponding column family. The Select Column Qualifier field is displayed as a drop down list. These values are retrieved based on the Sampling Range provided in the Schema tab.
Column Qualifier Type	The Select Column Qualifier field supports the following data types: <ul style="list-style-type: none"> ■ java.lang.String ■ byte array ■ java.lang.Object

Note:

If you change the values of **Table Name** and **Column Family** of **Schema** tab then, the following fields for **Columns** tab changes:

- The values of **Column Family** and **Select Column Qualifier** changes automatically.
- The **Column Family** changes automatically, but you need to manually change the values of **Enter Column Qualifier**.

Note:

You must select an appropriate data type for the Column Qualifiers, based on the data type being inserted.

8. Use the **WHERE** tab to specify the conditions for retrieving records:
 - a. Select the  icon to define new WHERE clause fields.
 - b. Select a logical operator from the **AND/OR** field, an Operator as needed, and specify values for the following fields:

Name	Description/Action
Parameter Key	Defines a unique key.

Name	Name Description/Action
	<p>The following are the two scenarios which defines the usage of parameter key:</p> <ul style="list-style-type: none"> ■ Rows in both tables in the WHERE clause will be inter dependent if it contains same parameter key. <p>For example, if you delete a row having parameter key as 1, then it automatically deletes the row having the same parameter key from the other table.</p> <ul style="list-style-type: none"> ■ Two rows having same key within a single table defines its relation, to achieve the functionality of IS/IS NOT operator. The behaviour of filters are decided based on the provided relation. <p>For example, if you have two rows with same parameter key within a table, then the filter is performed based on the respective IS/IS NOT operator.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: By default, the number of unique keys are between 1 to 25. To increase the number of unique keys, set the watt property as, <code>watt.hbase.filters.limit=30</code>.</p> </div>
Compound Operators	<p>Displays the following operators:</p> <ul style="list-style-type: none"> ■ AND ■ OR ■ SKIP ■ WHILE
Filter List	<p>Lists the following filters:</p> <ul style="list-style-type: none"> ■ FamilyFilter ■ FirstKeyOnlyFilter ■ InclusiveStopFilter ■ PageFilter ■ PrefixFilter ■ QualifierFilter ■ RowFilter ■ SingleColumnValueExcludeFilter ■ SingleColumnValueFilter

Name	Name Description/Action
	<ul style="list-style-type: none"> ■ ValueFilter ■ DependentColumnFilter
Family/Qualifier	Displays the list of Column Family/Column Qualifier/Column Family.Column Qualifier based on the selected Filter List .
Comparators	<p>Displays the list of following comparators based on the selected Filter List:</p> <ul style="list-style-type: none"> ■ binary ■ binaryprefix ■ regexstring ■ substring
Operator	<p>The following are the operators used:</p> <ul style="list-style-type: none"> ■ = ■ <= ■ != ■ > ■ < ■ >= ■ IS ■ IS NOT

Here are the ways to handle null values when passed as a parameter,

Operator	Input Field	Behaviour
IS	null	Filter with null in the input field will also accept NULL as an input and form the query.
	null T	Filter with null T accepts NULL as an input and fetches only the rows which passes the provided condition. It ignores the rows which does not have the provided qualifier.
	null F	Filter with null F accepts NULL as an input and fetches all the rows

Name	Name Description/Action		
	Operator	Input Field	Behaviour
			which passes the provided condition. It also fetches the rows which does not have the provided qualifier.
	IS NOT	null	Filter with null in the input field will not accept NULL as an input. If NULL is passed as a value, then corresponding filter will be ignored from the query.
		null T	Filter with null T will not accept NULL as an input. If NULL is passed as a value, then the corresponding filter will be ignored from the query. If any value other than NULL is passed, it fetches only the rows which passes the provided condition and ignores the rows which does not have the provided qualifier.
		null F	Filter with null F will not accept NULL as an input. If NULL is passed as a value, then the corresponding filter will be ignored from the query. If any value other than NULL is passed, it fetches all the rows which passes the provided condition and also fetches all the rows which does not have the provided qualifier.
Input Field Type	The data type of the Input Field . You can change this type if needed.		

Input Field The default value is ?, which acts as a placeholder for the variable so that you can set the input variable for that column at run time, or get input external to this adapter service. You can also type a fixed value in this field now or at run time. If you type a fixed value, be sure that it is valid, or an exception will be generated at run time.

The adapter automatically generates values for the following fields:

Field	Description/Action
Parameter Key	The unique key for those rows in the above table which takes input at runtime.

Field	Description/Action
Filters (second occurrence of this field)	The name of the column you need to use in the WHERE clause.
Input Field Type	The corresponding data type of an Input Field .
Input Field (second occurrence of this field)	The name of the input field. By default the name combines the values of Parameter and Field fields. However, you can also choose to specify any custom value.

- c. If necessary, use the or icons to change the order of the WHERE clause to ensure the parameters are parsed in the correct order.
- d. Repeat this procedure until you have specified all WHERE parameters.

Note:

If u do not pass any input values or if you explicitly pass the value as a NULL character, then the corresponding filters are skipped.

Note:

You must select an appropriate data type for the Column Qualifier filters and Value filters, based on the data type being inserted.

9. Use the **Configuration** tab to specify the configurations for the required records:
 - a. Select the  icon to define new Configuration fields.
 - b. Specify values for the following fields:

Name	Description/Action
Name	<p>Specifies the field in which the configuration properties are applied. The following are the list of available configuration properties:</p> <ul style="list-style-type: none"> ■ StartRow: Refers to RowID from which the service can start retrieving records. ■ StopRow: Refers to RowID till where the service can retrieve records. ■ Reverse: Requires input values as boolean. It can either be true or false. ■ RowPrefixFilter: Refers to retrieve the records for the mentioned RowID prefix.
Value	The input value provided for the field.

Name	Name Description/Action
Input Type	The data type of the Input Field . You can change this type if needed.
Input Field	The default value is ?, which acts as a placeholder for the variable so that you can set the input variable for that column at run time, or get input external to this adapter service. You can also type a fixed value in this field now or at run time. If you type a fixed value, be sure that it is valid, or an exception will be generated at run time.

10. To verify input or output information for the service, use the **Input/Output** tab as required.

11. From the **File** menu, select **Save**.

6 Predefined Health Indicator

■ Predefined Health Indicator	58
-------------------------------------	----

Predefined Health Indicator

Microservices Runtime includes predefined health indicators for some of its basic components. The health indicator captures the connection details for all the WmART based adapters at runtime. For more information, see *webMethods Adapter Runtime User's Guide*.

7 Administrator APIs

■ Administrator APIs	60
----------------------------	----

Administrator APIs

The Administrator APIs are available for Adapter for Apache HBase. For more information about Administrator APIs and samples, see *webMethods Adapter Runtime User's Guide*.

8 Configuration Variables Templates for Adapter Assets in Microservices Runtime

- [Configuration Variables Templates for Adapter Assets in Microservices Runtime](#) 62

Configuration Variables Templates for Adapter Assets in Microservices Runtime

The webMethods Adapter Runtime (ART) asset properties that can be configured from Integration Server Administrator are available in the configuration variables template (`application.properties` file) generated by Microservices Runtime. For more information, see *webMethods Adapter Runtime User's Guide* and *Developing Microservices with webMethods Microservices Runtime*.

9 Logging and Exception Handling

■ Overview of Logging and Exception Handling	64
■ Adapter for Apache HBase Logging Levels	64
■ Accessing Adapter Logging Levels	64
■ Changing Logging Levels	64
■ Adapter for Apache HBase Message Logging	64
■ Adapter for Apache HBase Error Codes	65

Overview of Logging and Exception Handling

The following sections describe Adapter for Apache HBase message logging, and exception handling. A list of error codes and supporting information appears at the end of this chapter.

Adapter for Apache HBase Logging Levels

Adapter for Apache HBase uses the Integration Server logging mechanism to log messages. You can configure and view the Integration Server logs to monitor and troubleshoot Adapter for Apache HBase. For detailed information about logging in Integration Server, including instructions for configuring and viewing the different kinds of logs supported by the server, see the *webMethods Integration Server Administrator's Guide* for your release.

Accessing Adapter Logging Levels

With Integration Server, you can configure different logging levels for Adapter for Apache HBase.

➤ **To access the adapter's logging information**

1. From the Integration Server Administrator screen, select **Settings > Logging**.

The **Logging Settings** screen appears. The **Loggers** section has **Adapters** included in the **Facility** section.

2. Expand the **Adapters** tree to see a list of all installed adapters with their code number and adapter description, along with the logging level.

Changing Logging Levels

You use Integration Server to change the logging levels.

➤ **To change logging settings for the adapter**

1. Click **Edit Logging Settings**. Select the required Level of **Logging** for Adapter for Apache HBase.
2. After making your changes, click **Save Changes**.

Adapter for Apache HBase Message Logging

Integration Server maintains several types of logs; however, Adapter for Apache HBase only logs messages to the audit, error and server logs. Because Adapter for Apache HBase works in conjunction with the WmART package, the adapter's messages and exceptions typically appear within log messages for the WmART package.

The logging levels for Adapter for Apache HBase are given in the following table.

Integration Server	Log	Description
Integration Server and higher	Audit Log	You can monitor individual adapter services using the audit log as you would audit any service in Integration Server. The audit properties for an adapter service are available in each Adapter for Apache HBase service template on the Audit tab.
	Error Log	Adapter for Apache HBase automatically posts fatal-level and error-level log messages to the server's error log. These log messages will appear as adapter run-time messages
	Server Log	Adapter for Apache HBase posts messages to the server log, depending on how the server log is configured. Fatal-level through debug-level log messages appear as adapter run-time log messages. Trace-level log messages appear as Adapter for Apache HBase log messages.

Adapter for Apache HBase's log messages appear in either of the following formats:

- ADA.506.nnnnc
- ADA.0506.nnnnc

where the facility code ADA indicates that the message is from an adapter, 0506 or 506 indicate that it is Adapter for Apache HBase, nnnn represents the error's minor code, and (optionally) crepresents the message's severity level. For detailed descriptions of Adapter for Apache HBase's minor codes, see [“Adapter for Apache HBase Error Codes” on page 65](#).

To monitor Adapter for Apache HBase's log messages in the server log, ensure that your server log's logging settings are configured to monitor the following facilities:

- 0113 Adapter Run time (Managed Object)
- 0114 Adapter Run time
- 0117 Adapter Run time (Adapter Service)
- 0118 Adapter Run time (Connection)

Adapter for Apache HBase Error Codes

The following table lists Adapter for Apache HBase's minor codes and provides information on the error message, reason, and possible action for each error.

Error Code	Description
1011	Please be sure that zookeeper master is running on the hostname and port number.
	Explanation: Client is not able to connect to zookeeper master.

Error Code	Description
	Action: Check if host and port entered in connection configuration is correct and reachable.
1012	Could not establish zookeeper connection on hostname and port number.
	Explanation: Zookeeper master is not running for the host mentioned in Connection configuration.
	Action: Check if zookeeper master is running on the mentioned host and port in connection configuration.
1015	Error occurred while trying to get resource domain values for service and resource domain.
	Explanation: Resource domain lookup failed for the indicated adapter service and resource domain.
	Action: Ensure that the connection parameters are valid, and then retry the operation.
1021	Scanner Timeout Exception Occurred.
	Explanation: Thrown when scanner is time-outs.
	Action: Increase the default value of 'hbase.client.scanner.timeout.period'
1023	Invalid input value for Timestamp field.
	Explanation: Value entered in the Timestamp field is not long data type.
	Action: Enter long data type values in Timestamp field.
1027	Row too big exception occurred while executing service.
	Explanation: Size of the data to be retrieved is greater than the configured block size.
	Action: Please increase the default block size and try increasing the value of client property 'hbase.client.scanner.max.result.size'
1029	Please enter correct Row Ids.
	Explanation: Row Id is not entered in the service.
	Action: Please enter the proper Row Id.
1031	Null pointer exception occurred. Please enter required input value.
	Explanation: Values are not entered in the Column Qualifiers.
	Action: Please enter the values for selected Column Qualifiers.
1032	No Column qualifier entered to insert/update the value.

Error Code Description

Explanation: No value is entered for selected Column Qualifier or no Column Qualifier is selected to insert/update.

Action: Enter proper value to selected Column Qualifier.

1033 Sampling range should be in proper format OR above 0.

Explanation: Entered sampling range is not of correct format.

Action: Enter the sampling range in the correct format which is ordered lexicographically.
