

webMethods Adapter for Terracotta Installation and User's Guide

Version 10.5

October 2019

This document applies to webMethods Adapter for Terracotta 10.5 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 © 2017-2021 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-TDA-IUG-105-20210422

Table of Contents

About this Guide.....	5
Document Conventions.....	6
Online Information and Support.....	7
Data Protection.....	7
 1 Overview of the Adapter.....	 9
About the Adapter.....	10
Terracotta Concepts.....	10
Architecture and Components.....	11
Package Management.....	12
Adapter Connections.....	12
Adapter Services.....	13
Viewing the Adapter's Update Level.....	17
Controlling Pagination.....	17
 2 Installing and Uninstalling Adapter for Terracotta.....	 19
Overview of installing and uninstalling Adapter for Terracotta.....	20
Requirements.....	20
The Integration Server Home Directory.....	20
Installing Adapter for Terracotta.....	20
Uninstalling Adapter for Terracotta.....	21
 3 Package Management.....	 23
Overview of Package Management.....	24
Adapter for Terracotta Package Management.....	24
Group Access Control.....	26
 4 Adapter for Terracotta Connections.....	 29
Overview of Adapter for Terracotta Connections.....	30
Before Configuring or Managing Adapter Connection.....	30
Configuring an Adapter for Terracotta Connection.....	30
 5 Adapter Services.....	 35
Overview of Adapter Services.....	36
Before Managing Adapter Services.....	36
Configuring Insert Service.....	36
Configuring Update Service.....	39
Configuring Delete Service.....	41
Configuring Query Service.....	42
 6 Terracotta Adapter Administrator APIs.....	 47
Adapter for Terracotta Administrator APIs.....	48

7 Managing DataSets.....	49
Overview of Managing Dataset Operations.....	50
Creating DataSets.....	50
Deleting Datasets.....	51
Viewing DataSets.....	51
 A Built-In Services.....	 53
Overview.....	54
wm.adapter.wmtcdb.cellIndexes:create.....	54
wm.adapter.wmtcdb.cellIndexes:get.....	54
wm.adapter.wmtcdb.cellIndexes:remove.....	55

About this Guide

- Document Conventions 6
- Online Information and Support 7
- Data Protection 7

This guide describes how to install, configure, and use the webMethods Adapter for Terracotta. This guide also describes the built-in services provided by the Adapter for Terracotta. It contains information for administrators and application developers who want to interact with Terracotta Store(TCStore) through the use of Terracotta Store Client API.

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

- The basic concepts and tasks for working with TCStore
- 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

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

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

About the Adapter

webMethods Adapter for Terracotta is an add-on to webMethods Integration Server that enables you to interact with Terracotta Store (TCStore) of Terracotta . Terracotta is a comprehensive, distributed in-memory data management solution which caters to caching and operational storage use cases, and enables transactional and analytical processing. Terracotta has one of the most powerful query and computation capabilities in its class, leveraging native JDK features such as Java Streams, collections, and functions. Terracotta supports the following sub-systems:

- TCStore - A storage sub-system that caters to operational store and compute functionality. The API exposing this sub-system's functionality is the TCStore API.
- Ehcache - A caching sub-system that caters to caching functionality. The API exposing this sub-system's functionality is the Ehcache API.

Both sub-systems are backed by the Terracotta Server, which provides a common platform for distributed in-memory data storage with scale-out, scale-up and high availability features.

Using Adapter for Terracotta, you can create and run adapter services to perform operations such as insert, update, delete and query data from a DataSet. The adapter also supports Administration operations to create, delete, and list dataset.

Terracotta Concepts

This section discusses the following concepts:

Dataset

Data is organized by Terracotta Store into collections called Datasets. Each dataset is comprised of zero or more records. Records are uniquely identified by a key within a dataset, but these records are not required to have uniform content. Each record can be comprised of cells having different names and/or different types.

Records

Instances in a dataset are called Records. Each record has a key, unique within a dataset, and zero or more cells. Record instances held within a given dataset are immutable. Changing one or more values in a record creates a new instance of that record which replaces the old instance.

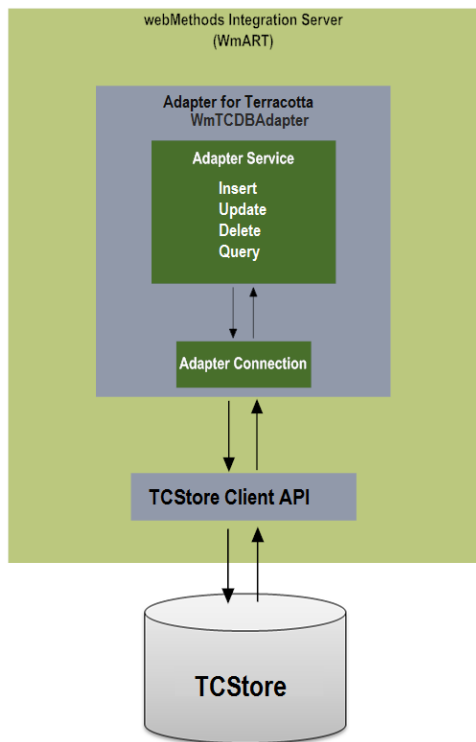
Cell Definitions and Values

A Cell Definition is a *type/name* pair. A record contains zero or more cell instances which are derived from a cell definition. You can create a cell to store in a record. The name of the cell is same as that of the cell definition and the value of the cell is of type specified in the cell definition while creating a cell.

Architecture and Components

Adapter for Terracotta provides a set of user interfaces and services that enable you to create integration with Terracotta store. 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 Terracotta” 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 Terracotta store's backend.



- **webMethods Integration Server.** Adapter for Terracotta is installed and runs on Integration Server.
- **(WmART).** The WmART package provides a common framework for webMethods Adapter for Terracotta version and later to use Integration Server's functionality, making Integration Server the run-time environment for Adapter for Terracotta. The WmART package is installed with Integration Server and it provides logging, error handling for the adapter, connections, and services.
- **Adapter for Terracotta.** The Adapter for Terracotta is delivered as a single package called WmTCDBAdapter. 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 Service.** Adapter services enable the Integration Server to initiate and perform database operations on TCStore. You configure adapter services using adapter services templates, which are provided with Adapter for Terracotta . For more information about adapter services, see [“Overview of Adapter Services” on page 36](#).
- **Adapter Connection.** Adapter connections enable the Integration Server to connect to TCStore 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 Terracotta Connections” on page 30](#).
- **TCStore Client API.** TCStore Client API is a client jar provided by TCStore that connects to the TCStore cluster.
- **TCStore.** TCStore is an Aggregate oriented, Key-Value NoSQL store. The individual values stored within TCStore contain cells with type information enabling the store to make use of the data it holds. TCStore is schema-less in its core design, allowing individual entries to contain identical sets of cells, a subset of common cells, or a completely different sets of cells. Unlike other NoSQL stores, TCStore does not use patterns that are traditional to tabular data or RDBMSs.

Package Management

Adapter for Terracotta is available as a package called WmTCDBAdapter. You can manage the WmTCDBAdapter 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 Terracotta 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 26](#).

Adapter Connections

Adapter for Terracotta connects to TCStore through Client jar API at run time. You create one or more connections at design time to use in integrations.

For instructions on configuring Adapter for Terracotta connections, see [“Overview of Adapter for Terracotta Connections” on page 30](#). 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 reuse 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 **Maximum 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 Terracotta Connection” on page 30](#).

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 Terracotta, 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 TCStore. The adapter services perform dataset operations by calling TCStore Client jar API. Integration Server then uses adapter connections that you defined earlier to execute the adapter services. For details, see [“Adapter Service Processing” on page 16](#).

Adapter services are based on templates provided with Adapter for Terracotta. Each template represents a specific operation on a resource, such as using the Insert service template to insert specified information to the dataset.

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 Terracotta provides the following adapter service templates:

Adapter Service Template	Description
Insert Service	<p>Inserts the record into a dataset and includes a mapping for an output field stores the result of service execution.</p> <p>For instructions about configuring the service, see “Configuring Insert Service” on page 36.</p>
Update Service	<p>Updates the existing record in a dataset like updating the value of an existing cell, inserting new cells or removing existing cells for the given Key and includes a mapping for an output field that stores the result of service execution.</p> <p>For instructions about configuring the service, see “Configuring Update Service” on page 39.</p>
Delete Service	<p>Deletes the record from a dataset and includes a mapping for an output field stores the result of service execution.</p> <p>For instructions about configuring the service, see “Configuring Delete Service” on page 41.</p>
Query Service	<p>Retrieves the records from a dataset.</p> <p>For instructions about configuring the service, see “Configuring Query Service” on page 42.</p>

Using Adapter Services

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

For this task...	Use these tools...
1. Create an adapter connection. For details, see “Overview of Adapter for Terracotta Connections” on page 30.	Integration Server Administrator
2. Select the appropriate adapter service template and configure the adapter service. Depending on the type of adapter service, you specify: <ul style="list-style-type: none"> ■ 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 36.	Designer
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.	Designer
4. Manage the adapter service.	Designer Integration Server

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 are provided in the WmART package's `pub.art.service` folder. Using the service `setAdapterServiceNodeConnection`, 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.](#)

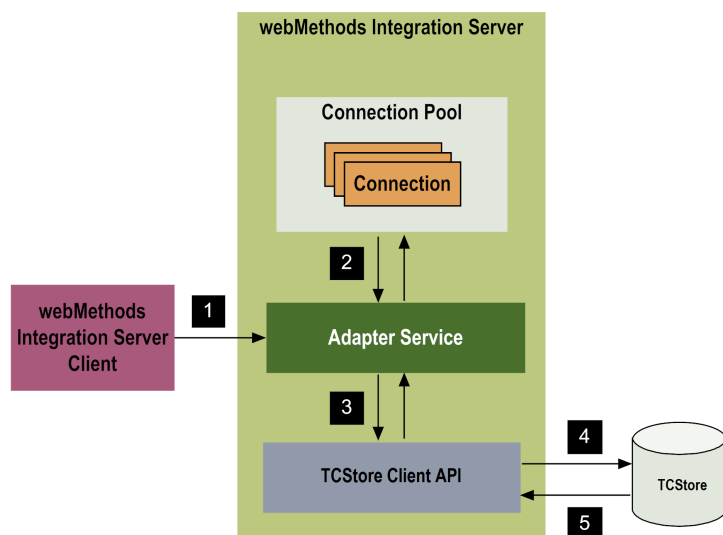
Change the Connection Associated with an Adapter Service at Run Time

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 Terracotta processes adapter services at run time.



Step	Description
1	An Integration Server client, typically using a flow or Java service, invokes a Adapter for Terracotta service on Integration Server to perform an operation on a TCMStore. 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 TCMStore.
3	The adapter service uses the TCMStore Client API to connect to TCMStore.
4	All adapter services performs the CRUD operation against a dataset in TCMStore. ■ For Insert, Delete, Upsert, and Query services, the adapter service performs the corresponding operation against the dataset in TCMStore.

Step	Description
5	<p>Depending on the adapter service type, such as a Query service, the adapter service may return data to Integration Server.</p> <ul style="list-style-type: none"> ■ If the operation is successful, the service returns the output from the Client jar API, if applicable. ■ If the operation is unsuccessful, the service returns an error such as an Adapter Exception. If the TCStore throws an exception while performing the adapter service's operation, the adapter logs the exception in the Integration Server log.

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 , 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 Terracotta

■ Overview of installing and uninstalling Adapter for Terracotta	20
■ Requirements	20
■ The Integration Server Home Directory	20
■ Installing Adapter for Terracotta	20
■ Uninstalling Adapter for Terracotta	21

Overview of installing and uninstalling Adapter for Terracotta

This chapter explains how to install, and uninstall webMethods Adapter for Terracotta . The instructions use the Software AG Installer and Software AG Uninstaller 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 Terracotta, see *webMethods Adapters System Requirements* .

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

Note:

Software AG strongly recommends that you configure your Terracotta cluster with the failover-priority set to consistency. If you set it to availability, you might lose data during outage scenarios

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 Terracotta

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 , choose the 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 for Terracotta** .

If you are using Integration Server , 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. After the installation completes, close the Installer and start the host Integration Server.

Note:

Be sure that TCStore Client libraries are installed while you are installing webMethods Adapter for Terracotta.

Uninstalling Adapter for Terracotta

➤ To uninstall Adapter for Terracotta

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 for Terracotta** . You can also choose to uninstall documentation.
4. After Uninstaller completes, restart the host Integration Server.

Note:

Uninstaller removes all Adapter for Terracotta 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 WmTCDBAdapter directory.

3 Package Management

■ Overview of Package Management	24
■ Adapter for Terracotta Package Management	24
■ Group Access Control	26

Overview of Package Management

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

Adapter for Terracotta Package Management

Adapter for Terracotta is provided as a package called WmTCDBAdapter. You can manage the WmTCDBAdapter 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 WmTCDBAdapter 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 WmTCDBAdapter package. That way, when the WmTCDBAdapter 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 26.

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, WmTCDBAdapter. (The WmTCDBAdapter 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 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 (WmTCDBAdapter).

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 (WmTCDBAdapter) 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.

4 Adapter for Terracotta Connections

■ Overview of Adapter for Terracotta Connections	30
■ Before Configuring or Managing Adapter Connection	30
■ Configuring an Adapter for Terracotta Connection	30

Overview of Adapter for Terracotta Connections

This chapter describes how to configure and manage Adapter for Terracotta 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 Terracotta on the same machine. For details, see [“Overview of installing and uninstalling Adapter for Terracotta” on page 20](#).
2. Make sure you have Integration Server administrator privileges so that you can access Adapter for Terracotta '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 WmTCDBAdapter 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 Terracotta Connection

When you configure Adapter for Terracotta , you specify information that Integration Server uses to connect to an Terracotta store. You can configure Adapter for Terracotta 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 Terracotta** .
2. In the Connections screen, click **Configure New Connection**.
3. In the Connection Types screen, click **webMethods Adapter for Terracotta Connection** to display the Configure Connection Type screen.
4. In the **webMethods Adapter for Terracotta** section, use the following fields:

Field	Description/Action
Package	<p>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.</p> <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 Terracotta, see “Adapter for Terracotta Package Management” on page 24.</p>
Folder Name	Specifies the folder in which you create the connection.
Connection 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.

5. In the **Connection Properties** section, use the below field:

Field	Description/Action
Terracotta URI	The URI which is passed takes the form, terracotta://server1:port,server2:port,...,serverN:port, where the URI stands for URI of the cluster.
Connection Timeout(sec)	Defines the timeout value in seconds, when you connect to Terracotta cluster the first time. The default value is 20 seconds.
ReConnect Timeout(sec)	Defines the timeout value in seconds to reconnect to the server, when the server connection disappears during the operation. The default value is 20 seconds.
DatasetCache Size	Specifies the number of datasets to be cached to improve the performance. The default value is 20. If you specify the value as 0, then no cache is created.
Secured Directory Path	Specify the path to security root directory to use the connection. For more information, refer the SSL/TLS Security Configuration section of Terracotta in <i>Terracotta Server Administration Guide</i> .

Note:
ReConnect Timeout(sec), DatasetCache Size, Secured Directory Path are connection properties specific to Terracotta 10.2.

6. 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	<p>If connection pooling is enabled, this field specifies the number of milliseconds that Integration Server will wait to obtain a connection 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.</p>
Expire Timeout	<p>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.</p> <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</p>

Field	Description/Action
	could have an adverse effect if your resource has a limited number of connections. 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.
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 value is 0.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

7. 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.

5 Adapter Services

■ Overview of Adapter Services	36
■ Before Managing Adapter Services	36
■ Configuring Insert Service	36
■ Configuring Update Service	39
■ Configuring Delete Service	41
■ Configuring Query Service	42

Overview of Adapter Services

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

Before Managing Adapter Services

Perform the following steps before configuring or managing adapter services.

➤ To prepare to configure or manage Adapter for Terracotta services

1. Start the Integration Server and Integration Server Administrator, if they are not already running.
2. Make sure you have administrator privileges to access the Adapter for Terracotta 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 WmTCDBAdapter package is enabled. For instructions, see [“Enabling Packages” on page 25](#).
4. Using Integration Server Administrator, configure an adapter connection with the adapter service. For instructions, see [“Overview of Adapter for Terracotta Connections” on page 30](#).

Note:

Integration Server provides built-in services that 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 must always define them in user-defined packages rather than in the WmTCDBAdapter package. For more information about managing packages for the adapter, see [“Overview of Package Management” on page 24](#).

Configuring Insert Service

An Insert service inserts a new record into the dataset. You configure Adapter for Terracotta services using Designer. For more information about adapter services, see [“Using Adapter Services” on page 15](#).

Read the section [“Before Managing Adapter Services” on page 36](#), before you configure adapter services.



➤ To configure an adapter service using the **Insert** template

1. In Designer, right-click the folder in which the service is to be contained and select **New > Adapter Service**.
2. Select the parent namespace, type a name for the adapter service, and click **Next**.
3. Select **webMethods Adapter for Terracotta** 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 **Insert** 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 **Dataset** tab to configure the following fields:

Field	Field Description/Action
Dataset Name	Specify the name of the dataset in TCStore.
Key Type	Specifies the key type which is auto populated based on the selected dataset.
Input Key Type	Specify the input key type.
Sampling Limit	<p>Specify the number of records to be scanned to the metadata of the cells. These cell definitions are listed in the Insert tab.</p> <p>The Input values are integers and it must be greater than 0.</p> <p>For example, if you provide sampling range as 10, then the service scans 10 records of the selected dataset and displays the cell definition for those 10 records.</p>
IS Document Reference	<p>Specify the IS Document namespace.</p> <p>The IS Document reference is of the format: <i>folder1.folder2:Documentname</i></p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: Each field in an IS Document is considered as an individual cell and the Cell Name is auto-populated.</p> </div>

7. Use the **Insert** tab to define the cell definitions using the following fields:
 - a. Use the  icon to create a new row as needed. You can use the  icon to fill in all cell definitions to the table that are retrieved after scanning the dataset for specified **Sampling**

Limit. Additionally, the second table will have the below fields with respect to the provided IS Document in the **Dataset** tab and the fields for the third table can be entered manually.

Use the following fields:

Field	Description/Action
Cell Name	Specifies the cell name in the record of a dataset.
Cell Type	Specifies the cell type in the record of a dataset. The supported data types are: INT, LONG, DOUBLE, STRING, CHAR, BOOL(Boolean), BYTES(byte array).
Input Field	Specifies the input field name.
Input Field Type	Specifies the data type of the input field.

The input variables are grouped under the document called **value**

Note:

Be sure to have unique cells across the tables in the **Insert** tab.

- 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
key	Specify the key of a Record.
value	Specifies the document which contains fields that needs to be inserted.
\$datasetName	Specify the dataset name. Optional field, where the specified value overrides the configured value in the Dataset tab.

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

Field	Description/Action
success	Specifies the result of the service. It can either be <i>true</i> or <i>false</i> .

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

To store the entire IS Document in a cell, perform the following:

- You can store the entire IS Document as a string in TCStore. For this, you have to select the Input Field Type as IData. When the Cell Type is a string and the Input Field Type is an IData,

the Adapter for Terracotta converts the IS Document to JSON string and stores in TCStore as a string.

- You can store the entire IS Document as Bytes in TCStore. For this, you have to select the Input Field Type as IData. When the Cell Type is a byte and the Input Field Type is an IData, the Adapter for Terracotta converts the IS Document to byte array and stores in TCStore as bytes.

Configuring Update Service

An Update service updates an existing record in a dataset and includes a mapping for output field that stores the result of service execution. You can configure Adapter for Terracotta services using Designer. For more information about adapter services, see [“Using Adapter Services” on page 15](#).

Read the section [“Before Managing Adapter Services” on page 36](#), before you configure adapter services.

➤ To configure an adapter service using the Update template

1. In Designer, right-click the folder in which the service is to be contained and select **New > Adapter Service**.
2. Select the parent namespace, type a name for the adapter service, and click **Next**.
3. Select **webMethods Adapter for Terracotta** 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 **Update** 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 **Dataset** tab to the following fields:

Field	Field Description/Action
Dataset Name	Specify the name of the dataset in TCStore.
Key Type	Specifies the key type which is auto populated based on the selected dataset.
Input Key Type	Specify the input key type.
Sampling Limit	Specify the number of records to be scanned to the metadata of the cells. These cell definitions are listed in the Update tab.

The Input values are integers and it should be greater than 0.

Field	Field Description/Action
	For example, if you provide sampling range as 10, then the service scans 10 records of the selected dataset and displays the cell definition for those 10 records.
IS Document Reference	Specify the IS Document namespace. The IS Document reference is of format: <i>folder1.folder2:Documentname</i>

7. Use the **Update** tab to define the cell definitions using the following fields:
- Use the  icon to create a new row as needed. You can use the  icon to fill in all cell definitions to the table that are retrieved after scanning the dataset for the specified **Sampling Limit**. Additionally, the second table will have the below fields with respect to the provided IS Document in the **Dataset** tab and the fields for the third table can be entered manually.

Use the following fields:

Field	Description/Action
Cell Name	Specifies the cell name in the record of a dataset.
Cell Type	Specifies the cell type in the record of a dataset. The supported data types are: INT, LONG, DOUBLE, STRING, CHAR, BOOL(Boolean), BYTES(byte array).
Input Field	Specifies the input field name.
Input Field Type	Specifies the data type of the input field.
Cell Operation	Specify the operation that needs to be performed on the cell. You can perform the following operations: <ul style="list-style-type: none"> ■ UPSERT: Inserts a new cell or updates the value in the existing cell. ■ DELETE: Deletes the existing cell.

Note:

Be sure to have unique cells across the tables in the **Update** tab.

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
key	Specify the key of a record.
value	Specifies the document which contains fields that needs to be updated.
\$datasetName	Specify the dataset name. Optional field, where the specified value overrides the configured value in the Dataset tab.

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

Field	Description/Action
success	Specifies the result of the service. It can either be <i>true</i> or <i>false</i> .

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

Configuring Delete Service

A Delete service deletes the record from a dataset for the specified key value. You can configure Adapter for Terracotta services using Designer. For more information about adapter services, see [“Using Adapter Services” on page 15](#).

Read the section [“Before Managing Adapter Services” on page 36](#), before you configure adapter services.

➤ To configure an adapter service using the Delete template

- In Designer, right-click the folder in which the service is to be contained and select **New > Adapter Service**.
- Select the parent namespace, type a name for the adapter service, and click **Next**.
- Select **webMethods Adapter for Terracotta** as the adapter type and click **Next**.
- Select the appropriate **Adapter Connection Name** and click **Next**.
- From the list of available templates, select the **Delete** 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 **Dataset** tab to configure the following fields:

Field	Field Description/Action
Dataset Name	Specify the name of the dataset in TCStore.
Key Type	Specifies the key type which is auto populated based on the selected dataset.
Input Key Type	Specify the key of a record.

7. 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
key	Specify the key of a record.
\$datasetName	Specify the dataset name. Optional field, where the specified value overrides the configured value in the Dataset tab.

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

Field	Description/Action
success	Specifies the result of the service. It can either be <i>true</i> or <i>false</i> .

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

Configuring Query Service

A Query service retrieves the record based on a given key. The service retrieves all the records from the dataset if you do not provide a key value. You can configure Adapter for Terracotta services using Designer. For more information about adapter services, see [“Using Adapter Services” on page 15](#).

Read the section [“Before Managing Adapter Services” on page 36](#), before you configure adapter services.

» To configure an adapter service using the Query template



1. In Designer, right-click the folder in which the service is to be contained and select **New > Adapter Service**.
2. Select the parent namespace, type a name for the adapter service, and click **Next**.
3. Select **webMethods Adapter for Terracotta** 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 **Query** 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 **Dataset** tab to configure the following fields:

Field	Field Description/Action
Dataset Name	Specify the name of the dataset in TCStore.
Sampling Limit	<p>Specify the number of records to be scanned to the metadata of the cells. These cell definitions are listed in the Query tab.</p> <p>The Input values are integers and it must be greater than 0.</p> <p>For example, if you provide sampling range as 10, then the service scans 10 records of the selected dataset and displays the cell definition for those 10 records.</p>
Key Type	Specifies the key type which is auto populated based on the selected dataset.
Input Key Type	Specify the input key type.

7. Use the **Select** tab to define the cells that you need to retrieve from the dataset using the following fields:
 - a. Use the  icon to create a new row as needed. You can use the  icon to fill in all cell definitions that are retrieved by scanning the dataset for the specified **Sampling Limit**. Additionally, the second table will have the below fields with respect to the provided IS Document in the **Dataset** tab and the fields for the third table can be entered manually.

Use the following fields:

Field	Description/Action
Cell Name	<p>Specifies the cell name in the record of a dataset.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note:</p> <p>If the Cell Name consist of (.), then it is considered as a separate field in an IS Document. All such selected cells are merged into an IS Document.</p> </div>
Cell Type	Specifies the cell type in the record of a dataset.

Field	Description/Action
	The supported data types are: INT, LONG, DOUBLE, STRING, CHAR, BOOL(Boolean), BYTES(byte array).
Output Field	Specifies the output field name.
Output Field Type	Specifies the data type of the output field. Note: You must select the appropriate output field type to avoid conversion overflow.
Maximum Records	Specify the maximum number of records to be retrieved from a dataset. Set the value as 0, if you want to retrieve all the records from the dataset. The default value is 10.
Note: Be sure to have unique cells across the tables in the Select tab.	


8. Use the **Filter** tab to filter records matching the specified conditions:

- a. Use the  icon to create a new row as needed.

Use the following fields to specify the conditions:

Field	Description/Action
Parameter	The filter parameter is a number which is automatically generated.
Cell Name	Specifies the name of the cell in the record of a dataset.
Relational Operator	Specifies the operation performed on a cell. The supported operations are: =, <, >, >=, <=, !=
Input Field	Specifies the input field name.
Cell Type	Specifies the type of the cell in the record of a dataset. The supported data types are: INT, LONG, DOUBLE, STRING, CHAR, BOOL(Boolean), BYTES(byte array).
Input Field Type	Specifies the data type of the input field. Note: The result is not guaranteed for any operators along with =(equals) for the Date input field type.

Field	Description/Action
Filter Condition Name	Specifies the name of the filter condition .

9. Use the **Operands** tab to define aggregate filter conditions using the following fields:
- Use the  icon to create a new row as needed.

Use the following fields:

Field	Description/Action
Filter Condition Name	Specifies the name given for the filter condition which is provided in the Filter tab.
AND/OR	Specifies the logical operator.
Result	Specifies the result of the logical operation.
	Note: The resulting value can be used as an operand for other filter conditions.

For example, to use an aggregate filter condition such as ((Cond1 && Cond2) && (Cond5 && Cond6)) && (Cond3 || Cond4), where cond1 to cond6 are the filter condition names defined in the **Filter** tab can be represented as shown below:

```
Cond1 and Cond2 = result1
Cond3 or Cond4 = result2
Cond5 and Cond6 = result3
result1 and result3 = result4
result4 or result2 = result6
```

10. 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
key (Optional)	Specify the key of the Record.
value	Specifies the document which contains fields that needs to be queried.
\$datasetName	Specify the dataset name. Optional field, where the specified value overrides the configured value in the Dataset tab.

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

Field	Description/Action
results	Specifies a list of key value pairs, where the value contains list of cell names versus cell values .

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

This service displays the result as a list of key value pairs where the value is a document which contains the cell names with their corresponding values.

Note:

You can retrieve the cells with similar names but with different data types from a dataset by providing different Output Field names.

To retrieve the entire IS Document in a cell, perform the following:

- You can retrieve the entire IS Document that is stored as a string in TCStore. For this, you have to select the Output Field Type as IData. When the Cell Type is a string and the Output Field Type is an IData, the Adapter for Terracotta converts the JSON string to IS Document.
- You can retrieve the entire IS Document that is stored as bytes in TCStore. For this, you have to select the Output Field Type as IData. When the cell Type is bytes and the Output Field Type is an IData, the Adapter for Terracotta converts the bytes to IS Document.

Note:

The execution of Adapter service can fail with `AdapterConnectionException`, if a server connection is lost or refreshed when the TCDB server is restarted. In such circumstances, a retry of service execution or inspection of the state on reconnecting is required to determine the effect of the abandoned operation.

6 Terracotta Adapter Administrator APIs

■ Adapter for Terracotta Administrator APIs	48
---------------------------------------------------	----

Adapter for Terracotta Administrator APIs

The administrator APIs are available for Adapter for Terracotta. For more information, see *webMethods Adapter Runtime User's Guide*.

Sample Template for creating Connections:

```
{
  "connectionAlias": "Connection Alias",
  "adapterTypeName": "com.wm.adapter.wmtcdb.TCDBAdapter",
  "connectionFactoryType": "com.wm.adapter.wmtcdb.connection.TCDBConnectionFactory",
  "packageName": "Package Name",
  "connectionSettings": {
    "uri": "terracotta://localhost:9410",
    "connectionTimeout": "20",
    "secureDirPath": "",
    "reconnectTimeout": "20",
    "datasetCacheSize": "20"
  },
  "connectionManagerSettings": {
    "poolable": "true",
    "minimumPoolSize": "1",
    "maximumPoolSize": "10",
    "poolIncrementSize": "1",
    "blockingTimeout": "1000",
    "expireTimeout": "1000",
    "startupRetryCount": "0",
    "startupBackoffSecs": "10"
  }
}
```


7 Managing DataSets

■ Overview of Managing Dataset Operations	50
■ Creating DataSets	50
■ Deleting Datasets	51
■ Viewing DataSets	51

Overview of Managing Dataset Operations

This chapter describes how to create, view, and delete datasets in TCStore. For more details on datasets, refer *Terracotta documentation*.

Creating DataSets

You can manage the dataset operations manually on TCStore using the Integration Server Administrator screen.

➤ **To create a dataset**

1. In the **Adapters** menu of Integration Server Administrator's navigation area, click **webMethods Adapter for Terracotta**.
2. On the adapter's screen, click **Dataset Operations**.
3. In the **Dataset Admin Operations** section, use the following fields:

Field	Description/Action
DataSet Operation	Select the Create operation.
Connection Name	Select the configured connection.
DataSet Name	Specify the name of the dataset.
Key Type	Specify the key type of the dataset
Off-heap Resource	Specify the name that matches the name provided in the path <code>service/offheap-resources/resource</code> element of TCStore server XML configuration. Note: If you provide any invalid value for Off-heap resource then TCStore server shuts down and the Create operation responds after the timeout(15 secs).
Disk Resource	Specify the name that matches the name provided in the path <code>service/data-directories/directory</code> element of TCStore server XML configuration.

4. Click **Submit**.

Deleting Datasets

You can delete the dataset operations manually on TCStore using the Integration Server Administrator screen.

> To delete a dataset

1. In the **Adapters** menu of Integration Server Administrator's navigation area, click **webMethods Adapter for Terracotta**.
2. On the adapter's screen, click **Dataset Operations**.
3. In the **Dataset Admin Operations** section, use the following fields:

Field	Description/Action
DataSet Operation	Select the Delete operation.
Connection Name	Select the configured connection.
DataSet Name	Specify the name of the dataset.

4. Click **Submit**.

Viewing DataSets

You can view the dataset operations manually on TCStore using the Integration Server Administrator screen.

> To view a dataset

1. In the **Adapters** menu of Integration Server Administrator's navigation area, click **webMethods Adapter for Terracotta**.
2. On the adapter's screen, click **Dataset Operations**.
3. In the **Dataset Admin Operations** section, use the following fields:

Field	Description/Action
DataSet Operation	Select the Get operation.
Connection Name	Select the configured connection.

4. Click **Submit**.

A Built-In Services

■ Overview	54
■ <code>wm.adapter.wmtcdb.cellIndexes:create</code>	54
■ <code>wm.adapter.wmtcdb.cellIndexes:get</code>	54
■ <code>wm.adapter.wmtcdb.cellIndexes:remove</code>	55

Overview

This appendix provides information on the built-in services provided by webMethods Adapter for Terracotta. These services are located in the WmTCDBAdapter package.

wm.adapter.wmtcdb.cellIndexes:create

The `wm.adapter.wmtcdb.cellIndexes:create` Java service allows you to create cell indexes on the dataset's cells.

Input Parameters

<i>connectionName</i>	String. Required. The name of the connection.
<i>datasetName</i>	String. Required. The name of the dataset in which all the cell indexes are created.
<i>cellDefinitions</i>	Document. The list of cells(name and type) for which indexes are created.

Output Parameters

<i>result</i>	The result of the service. It can be either <i>true</i> or <i>false</i> .
---------------	---------------------------------------------------------------------------

wm.adapter.wmtcdb.cellIndexes:get

The `wm.adapter.wmtcdb.cellIndexes:get` Java service allows you to retrieve all the cell indexes that exist in the dataset.

Input Parameters

<i>connectionName</i>	String. Required. The name of the connection.
<i>datasetName</i>	String. Required. The name of the dataset from which all the cell indexes are retrieved.

Output Parameters

<i>result</i>	The list of indexes present in the dataset with cellname, celltype and index status.
---------------	--------------------------------------------------------------------------------------

wm.adapter.wmtcdb.cellIndexes:remove

The `wm.adapter.wmtcdb.cellIndexes:remove` allows you to delete the cell indexes that exist in the dataset.

Input Parameters

<i>connectionName</i>	String. Required. The name of the connection.
<i>datasetName</i>	String. Required. The name of the dataset from which all the cell indexes should be removed.
<i>cellNames</i>	String List. List of cell names whose indexes should be removed.

Output Parameters

<i>result</i>	The result of the service. It can be either <i>true</i> or <i>false</i> .
---------------	---------------------------------------------------------------------------
