

webMethods Remedy Adapter Installation and User's Guide

Version 7.1

July 2012

This document applies to webMethods Remedy Adapter 7.1 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 © 2005-2022 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-REM-IUG-71-20221202

Table of Contents

About this Guide.....	5
Document Conventions.....	6
Online Information and Support.....	7
Data Protection.....	8
 1 Overview of webMethods Remedy Adapter.....	 9
About the Adapter.....	10
Architecture and Components.....	10
Adapter Package Management.....	15
Adapter Connections.....	16
Adapter Services.....	18
Using Version Control Systems to Manage Adapter Elements.....	24
Optimize Infrastructure Data Collector Support for the Adapter.....	24
Viewing the Adapter's Update Level.....	24
Controlling Pagination.....	24
 2 Installing webMethods Remedy Adapter.....	 27
Overview.....	28
Requirements.....	28
Installing Remedy Adapter 7.1.....	28
Upgrading from Remedy Adapter 6.0.....	29
Uninstalling Remedy Adapter 7.1.....	30
 3 Adapter Package Management.....	 31
Overview.....	32
Managing the Adapter Package.....	32
Controlling Group Access.....	35
Using Remedy Adapter in a Clustered Environment.....	36
 4 Adapter Connections.....	 41
Overview.....	42
Before Configuring or Managing Adapter Connections.....	42
Configuring Adapter Connections.....	42
Dynamically Changing a Service's Connection at Run Time.....	45
Viewing Adapter Connection Parameters.....	45
Editing Adapter Connections.....	47
Copying Adapter Connections.....	48
Deleting Adapter Connections.....	48
Enabling Adapter Connections.....	49
Disabling Adapter Connections.....	49
 5 Adapter Services.....	 51

Overview.....	52
Before Configuring or Managing Adapter Services.....	52
Configuring Create Services.....	53
Configuring Delete Services.....	55
Configuring Find Services.....	56
Configuring Get Services.....	58
Configuring Batch Get Services.....	60
Configuring Update Services.....	62
Configuring ARS Server Operations Services.....	64
Testing Adapter Services.....	67
Viewing Adapter Services.....	67
Editing Adapter Services.....	67
Deleting Adapter Services.....	68
Validating Adapter Service Values.....	69
Reloading Adapter Values.....	70
 6 Configuring Remedy to webMethods Notifications.....	73
Overview.....	74
Setting Up E-mail Notifications.....	74
Setting Up Web Services.....	78
 7 Predefined Health Indicator.....	81
Predefined Health Indicator.....	82
 8 Administrator APIs.....	83
Administrator APIs.....	84
 9 Configuration Variables Templates for Adapter Assets in Microservices Runtime.....	85
Configuration Variables Templates for Adapter Assets in Microservices Runtime.....	86
 10 Adapter Logging and Exception Handling.....	87
Overview.....	88
Adapter Logging Levels.....	88
Configuring Adapter Logging Levels.....	88
Adapter Message Logging.....	89
Adapter Exception Handling.....	90
Adapter Error Codes.....	92
 A Data Mapping.....	99
Overview.....	100
AR System Data Type to Adapter Data Type Mapping.....	100
Unsupported Remedy Data Types.....	101
 B webMethods Remedy Adapter Services.....	103
Overview.....	104
WmRemedyAdapter Package.....	104

About this Guide

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

This guide describes how to configure and use webMethods Remedy Adapter. It contains information for administrators who configure and manage webMethods Integration Server and application developers who want to create applications that update objects in the Remedy Action Request System (AR System) and/or receive notification of updates to objects in the AR System.

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

- How to create flow, Java, and/or C/C++ services
- Terminology and basic operations of your operating system
- The basic concepts of the Remedy Action Request System
- The setup and operation of webMethods Integration Server.
- How to perform basic tasks with 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 webMethods Remedy Adapter

■ About the Adapter	10
■ Architecture and Components	10
■ Adapter Package Management	15
■ Adapter Connections	16
■ Adapter Services	18
■ Using Version Control Systems to Manage Adapter Elements	24
■ Optimize Infrastructure Data Collector Support for the Adapter	24
■ Viewing the Adapter's Update Level	24
■ Controlling Pagination	24

About the Adapter

webMethods Remedy Adapter (Remedy Adapter) is an add-on to webMethods Integration Server that enables you to exchange data with Remedy Action Request System (AR System), which is the base for all Remedy applications. The adapter provides seamless and real-time communication with the AR System Server. You can create integrations that initiate actions from webMethods Integration Server that are performed in the AR System, and you can detect changes in the AR System and send notification to Integration Server:

- Initiating action from Integration Server.

You can issue requests to the AR System to create, get, update, find, retrieve details, execute one of several adapter ARS Server operators, or delete entries in the AR System. Remedy Adapter provides user interfaces in the Integration Server Administrator that enable you to configure and manage adapter connections to the AR System Server. Remedy Adapter also provides user interfaces in Software AG Designer that you use to create adapter services. The adapter services use the connections you define to perform an action (create, get, update, find, retrieve details, execute one of several adapter ARS Server operators, or delete entries). To perform the action, Remedy Adapter uses the AR System Java APIs.

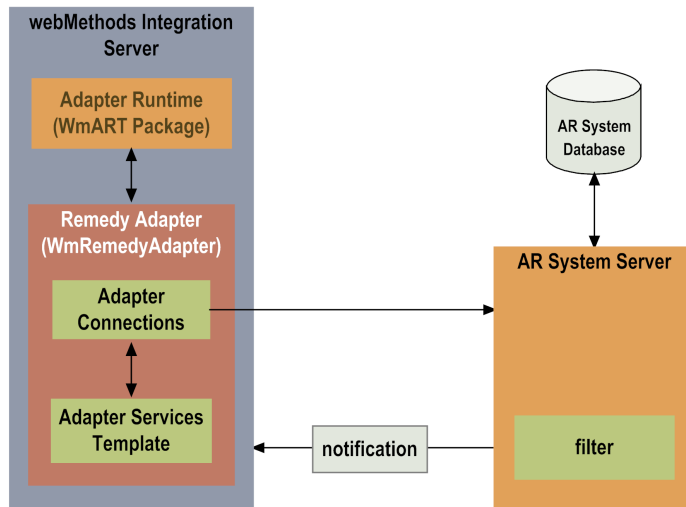
- Detecting changes in the AR System.

You can set up filters in the AR System that detect specific changes that occur in the AR System. When a change is detected, the AR System Server sends the information defined in the filter. To notify Integration Server of a change, you can set up the filter to send an e-mail message to Integration Server, or you can invoke a Web service.

Architecture and Components

Remedy Adapter provides a set of user interfaces, services, and templates that enable you to create integrations that interact with the AR System. The adapter is provided as a single package that must be installed on Integration Server. For detailed installation instructions, see [“Installing webMethods Remedy Adapter” on page 27](#). For software requirements, see the *webMethods Adapters System Requirements*.

The following diagram shows, at a high level, how the adapter components connect to an AR System Server:



■ Integration Server

Remedy Adapter is installed and runs on Integration Server.

■ Adapter Runtime (WmART Package)

The Adapter Runtime, which is provided in the WmART package, provides a common framework for webMethods 7 and later adapters to use the Integration Server's functionality, making Integration Server the run-time environment for Remedy Adapter. The WmART package is installed with Integration Server and provides logging, transaction management, and error handling for the adapter and its connections and services.

■ Remedy Adapter

Remedy Adapter is delivered as a single package called WmRemedyAdapter. The adapter provides administrative screens that enable you to configure and manage adapter connections. You access these administrative screens via Integration Server Administrator. Remedy Adapter also provides Designer user interfaces that enable you to configure and manage adapter services. Remedy Adapter installation includes templates from which all Remedy Adapter services can be created.

■ Adapter Connections

An adapter connection enables Integration Server to connect to the AR System Server at run time. You must configure an adapter connection before you can create adapter services. For a detailed description of adapter connections and usage information, see [“Adapter Connections” on page 16](#).

■ Adapter Service Templates

Use adapter services for transactions where Integration Server initiates a request in the AR System. An adapter service enables Integration Server to interact with the AR System Server to create, get, update, find, retrieve details, execute one of several adapter ARS Server operators, or delete entries in the AR System database.

Each entry in the AR System database uses a specific form. When you create an adapter service, you select an existing form to provide the schema for the entry upon which the adapter service is to act.

These adapter services act as a client to the AR System Server and use the AR SystemJava APIs to perform the requested actions. Remedy Adapter provides adapter service templates in Designer. For more information about adapter service templates and how the services interact, see [“Requests Initiated on Integration Server ” on page 12](#), [“Adapter Services” on page 18](#), and [“Adapter Services” on page 51](#).

- **AR System Server**

Remedy Adapter connections and services interact with the AR System Server. The AR System Server performs access control to the AR System and controls the flow of data into and out of the AR System database. For example, if you invoke an adapter service to create a new entry in the AR System, the AR System Server performs the action to update the database with the new entry.

- **AR System database**

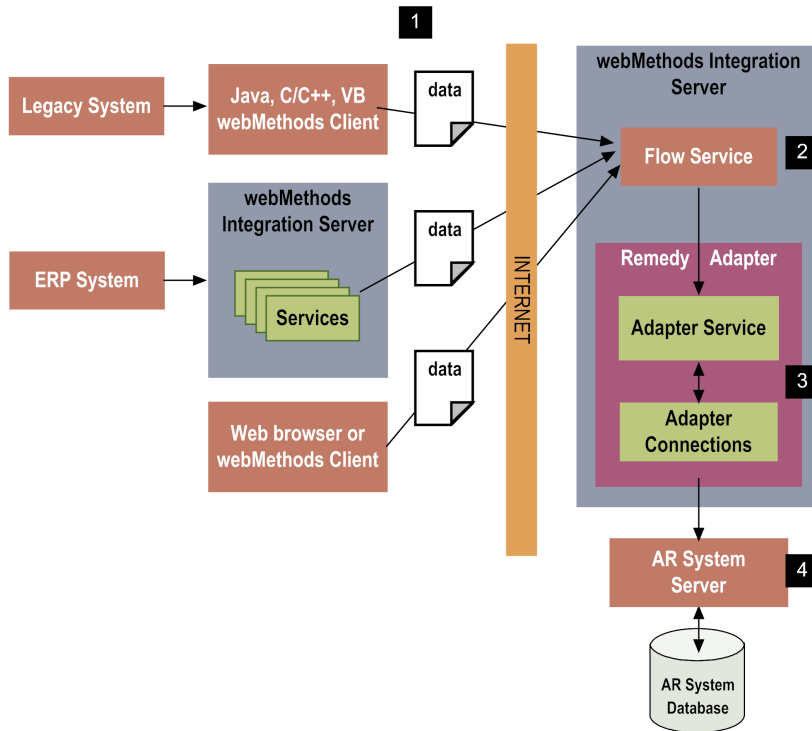
The data (entries) that the AR System maintains is stored in the AR System database.

- **Filters**

Use filters to have the AR System notify Integration Server of actions that occurs in the AR System. Filters are a feature of the AR System and are *not* provided by Remedy Adapter. You define filters in the AR System, and they are executed in the AR System Server. When you define a filter, you specify the condition that the filter is to detect and the action to take when that condition is met. For integrations with Integration Server, the action you specify is to notify Integration Server either by sending an e-mail message or by invoking a Web service. For more information, see the Remedy Adapter documentation.

Requests Initiated on Integration Server

The following diagram illustrates the use of Remedy Adapter and Integration Server in a typical business-process integration to initiate a request from Integration Server that is performed in the AR System. See the table below the diagram for additional information.



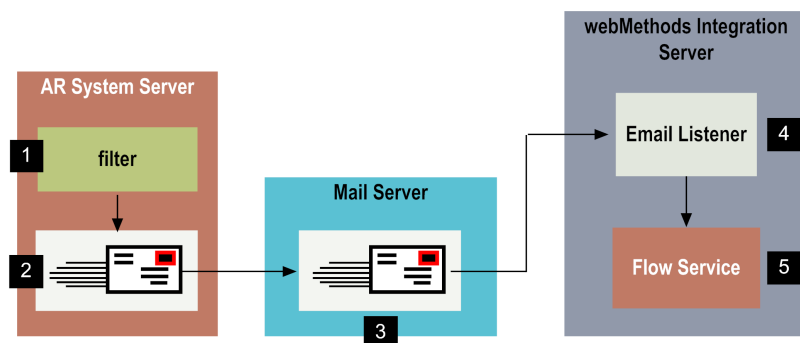
Step	Description
1	To request that an action be performed in the AR System, your back-end systems invoke services that you create and that reside in Integration Server. When invoking the service, Integration Server client sends data that provides more information about the request. For example, if the Integration Server client wants to create an entry in the AR System, the data would be the values to use in the entry to create.
2	The service that the Integration Server client invokes is executed on Integration Server. The service invokes adapter services to interact with the AR System.
3	The adapter services use an adapter connection to connect to the AR System Server. The adapter service interacts with the AR System via the AR System Java APIs.
4	The AR System Server receives the request and performs the request that the Integration Server client initiated.

Notifications Sent from the AR Systems

This section describes how Integration Server can receive notifications when an action occurs in the AR System. The AR System can send notifications either using an e-mail message or by invoking a Web service.

Receiving an E-mail Notification

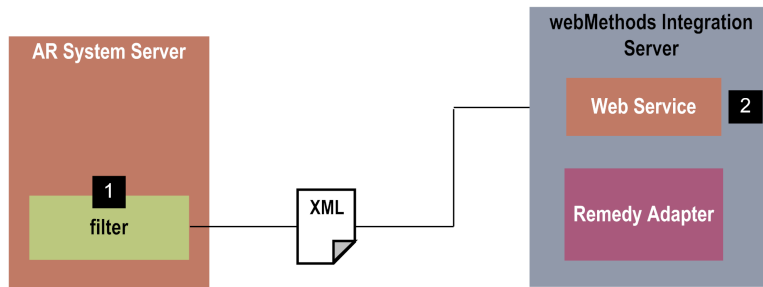
The following diagram illustrates how Integration Server can receive a notification in an e-mail message when an action occurs within the AR System. See the table below the diagram for additional information.



Step	Action
1	An action occurs in the AR System, and a filter that you created in the AR System detects this condition. The action defined in the filter is to notify via an e-mail message. As a result, the AR System Server sends an e-mail notification to the outgoing mailbox identified in the filter.
2	The outgoing mailbox sends the e-mail message to the e-mail account specified when the outgoing mailbox was configured.
3	The e-mail message is sent to a mail server that receives e-mail messages from AR System Server. The Integration Server Email Listener periodically checks the e-mail account on the mail server and polls for e-mail messages.
4	The Email Listener receives the e-mail message.
5	The Email Listener invokes the service that you defined to handle the notification. Remedy Adapter provides a sample service, <code>wm.adapter.wmremedy.outbound.emailNotification:receiveEmail</code> service, which illustrates how to obtain the content from the e-mail message.

Handling Notifications via a Web Service

The following diagram illustrates how the AR System can invoke a Web service to notify Integration Server when an action occurs within the AR System. See the table below the diagram for additional information.



Step	Action
1	An action occurs in the AR System, and a filter that you created in the AR System detects this condition. The action defined in the filter is to invoke a Web service. As a result, the AR System Server invokes the Web service and passes an XML document that contains the notification.
2	The Web service that the AR System Server invokes resides on the Integration Server that handles the notification. Remedy Adapter provides a sample Web service that you can use to model your own Web service. The sample Web service is <code>wm.adapter.wmremedy.outbound.webService.receiveWSDL_SOAP_RPC</code> .

Adapter Package Management

Remedy Adapter is provided as a package called `WmRemedyAdapter` that you manage like any package on Integration Server.

There are several considerations regarding how you set up and effectively manage your packages on Integration Server, such as those described in the following list.

- Configure user-defined packages for your adapter connections and adapter services. See [“Managing the Adapter Package” on page 32](#) for details.
- Understand how package dependencies work so you make the best decisions regarding how you manage your adapter services. See [“Package Dependency Requirements and Guidelines” on page 33](#) for details.
- Control which development groups have access to which adapter services. See [“Controlling Group Access” on page 35](#) for details.
- Understand how clustering, an advanced feature of webMethods Integration Server, works to effectively manage your adapter services. See [“Using Remedy Adapter in a Clustered Environment” on page 36](#) for details.
- Enable and disable packages. See [“Enabling Packages” on page 33](#) and [“Disabling Packages” on page 34](#) for details.
- Load, reload, and unload packages. See [“Loading, Reloading, and Unloading Packages” on page 34](#).

Adapter Connections

An adapter connection enables a Remedy Adapter service to connect to the AR System Server to create, update, find, get, retrieve details, execute one of several adapter ARS Server operators, and delete entries that are stored in the AR System database.

You can configure one or more adapter connections at design time to use in integrations. You configure one adapter connection for each AR System Server with which you want to integrate. When you define the adapter connection, you specify parameters, such as:

- Name for the connection
- AR System Server host name
- User name and password of a user defined on the AR System Server that Integration Server uses to establish the connection and to obtain information about forms, fields, etc.

You configure adapter connections using the Integration Server Administrator. You must have webMethods administrator privileges to access the Remedy Adapter's administrative screens.

For instructions for configuring, viewing, editing, enabling, disabling, and deleting Remedy Adapter connections, see [“Adapter Connections” on page 41](#). For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.

For a list of tasks that you must do before you can create your connections, see [“Before Configuring or Managing Adapter Connections” on page 42](#).

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 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** parameter. 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 **Pool Increment Size**) and adds them to the connection pool. If the pool is full (as specified in **Maximum Pool Size**), the requesting service will wait for Integration Server to obtain a connection, up to the length of time specified in the **Block Timeout** parameter, 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 **Expire Timeout**. You can enable the system to retry the initialization any number of times, at specified intervals.

For information about configuring connections, see [“Adapter Connections” on page 41](#).

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 built-in service `setAdapterServiceNodeConnection` enables you to change the connection associated with an adapter service. For more information, see the sections below.

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

Changing the Connection Associated with an Adapter Service at Design Time

Integration Server provides a built-in service that you can use at design time to change the connection associated with an adapter service. This built-in service is named `setAdapterServiceNodeConnection` and is 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 do not need to create and maintain multiple adapter services.

Note:

This built-in service can be run at design time only; do not use it within an Integration Server flow or Java service. You must run this service directly from Designer by selecting the 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 17](#).

Changing the Connection Associated with an Adapter Service at Run Time

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

For example, you can configure an adapter service to use a default connection that interacts with your company's production AR System Server. However, at run time you can override the default connection and instead use another connection to interact with the company's AR System Server.

For more information about overriding a service's default connection at run time, see [“Dynamically Changing a Service's Connection at Run Time” on page 45](#).

Adapter Services

Adapter services allow you to connect to the AR System Server and initiate an operation on the AR System from Integration Server. Integration Server uses adapter connections that you defined earlier to execute the adapter services. You invoke adapter services from flow or Java services to interact with the AR System Server to create, get, update, find, retrieve details, execute one of several adapter ARS Server operators, or delete entries in the AR System database. The adapter services perform these operations (for example, create, get, update, find, or delete) using the AR System Java APIs. Remedy Adapter makes synchronous calls to the AR System Server using the AR System Java APIs.

Entries in the AR System database are associated with specific AR System forms. A form defines the fields associated with an entry. At design time you configure an adapter service using templates provided with Remedy Adapter. When you configure an adapter service, you provide the form name. An adapter service template contains all the code necessary for accessing forms and obtaining fields. Each template represents a specific operation for doing work on an AR System Server, such as using the Get template to retrieve an entry from the AR System database.

You use Designer to configure the adapter service. For more information, see the *webMethods Service Development Help* for your release.

The input and output signatures for an adapter service are defined for you when you configure the adapter service. The input and output signatures depend on the function of the adapter service (for example, create, get, update, find, or delete) and the specific form that the adapter service is to use. For example, if you use the Create template, the input signature includes the fields from the form that you select when you configure a Create service. Similarly, if you use the Get template, the output signature includes the fields from the form that you select when you configure a Get service.

Adapter Service Templates

The following table lists the adapter service templates:

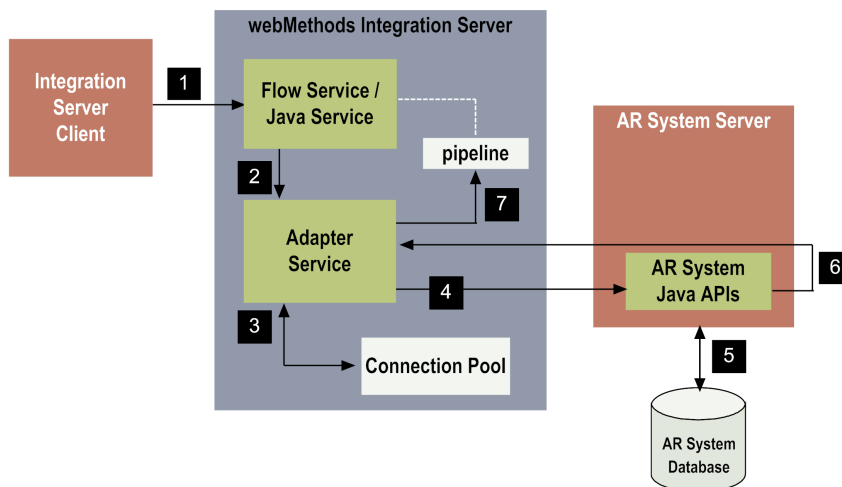
Adapter Service Template	Description
Create	Creates an entry in the AR System database based on a selected AR System form.
Delete	Deletes an entry with a specified entry ID number from the AR System database.
Find	Locates one or more entries that are stored in the AR System database and that match the specified search criteria.
Get	Retrieves the details for an existing entry in the AR System database that uses a specified form and has a specified entry ID.

Adapter Service Template	Description
Batch Get	Finds and retrieves the details for all existing entries in the AR System database that use a specified form and have the specified list of entry IDs.
Update	Updates fields for an existing entry in the AR System database.
ARS Server Operations	Executes one of the following ARS Server operations: <ul style="list-style-type: none"> ■ Execute Process ■ Get Operation Time ■ Begin Bulk Entry Transaction ■ End Bulk Entry Transaction

Adapter Service Operation

The run time actions that occur in Integration Server and the AR System when an adapter service is invoked are basically the same for all types of Remydy Adapter services. The only difference is the operation performed in the AR System.

At run time, an Integration Server client invokes a flow or Java service on Integration Server. This flow or Java service, in turn, invokes the adapter service. The adapter service uses AR System Java APIs to interact with the AR System Server. The diagram and table below illustrate the run-time processing for an adapter service.



Step	Description
1	An Integration Server client invokes a flow service or Java service on Integration Server.

Step	Description
2	The flow or Java service invokes the adapter service, which you configured earlier using Designer.
3	The adapter service gets a connection from the service's connection pool, which you created and enabled earlier using Integration Server Administrator.
4	Through its connection, the adapter service interacts with the AR System Server using the AR System Java APIs to perform the operation requested by the adapter service.
5	<p>The AR System Server performs the requested operation, interacting with the AR System database as necessary. For more information about a specific operation, see one of the following sections:</p> <ul style="list-style-type: none">■ “Create Adapter Service” on page 20■ “Delete Adapter Service” on page 21■ “Find Adapter Service” on page 21■ “Get Adapter Service” on page 21■ “Batch Get Adapter Service” on page 22■ “Update Adapter Service” on page 22■ “ARS Server Operations Adapter Service” on page 22
6	The AR System Server returns the result of the operation. See the appropriate section below for information about the outcome for a specific operation.
7	The adapter service updates results of the operation in the pipeline.

Create Adapter Service

Services configured with the Create template create entries in the AR System database.

At design time, you identify the AR System form that the Create adapter service is to use. Based on the form you select, Remedy Adapter loads the appropriate fields into the adapter service template. You select the fields that are needed for the entries that the service will create. The input signature of the Create adapter service is based on the form fields that you select. The output signature of the Create adapter service is the entry ID of the newly created entry; you cannot modify the output signature.

At run time, the Create adapter service uses the AR System Java APIs to create a new entry in the AR System database. The new entry has the format that is defined by an AR System form, which you identified when you configured the Create adapter service. The form identifies the fields that make up the entry. The Create adapter service retrieves the values for the fields from the pipeline.

If the Create operation is successful, the adapter service returns a String that contains the entry ID for the newly created entry. If the operation is unsuccessful, the adapter service throws an

AdapterException or AdapterConnectionException. For more information about how the adapter handles exceptions, see [Adapter Logging and Exception Handling](#).

Delete Adapter Service

Services configured with the Delete template delete an entry from the AR System database. The entry to delete is identified by the name of the AR System form for the entry along with the entry ID for the entry.

At design time, you identify the AR System form that the Delete adapter service is to use. You can also optionally supply the entry ID of the entry to delete. Alternatively, the entry ID can be supplied at run time. The output signature of the Delete adapter service contains a String that holds the status (or outcome) of the delete operation (Success or Failure); you cannot modify the output signature.

At run time, the Delete adapter service uses the AR System Java APIs to delete the entry specified by the form name and entry ID.

If the Delete operation is successful, the adapter service returns a String. If the operation is unsuccessful, the adapter service throws an AdapterException or AdapterConnectionException. For more information about how the adapter handles exceptions, see [Adapter Logging and Exception Handling](#).

Find Adapter Service

Services configured with the Find template locate one or more existing entries in the AR System database. The search criteria for locating entries are the name of the form used for the entries and a query string.

At design time, you identify the AR System form that the Find adapter service is to use. You can also optionally supply a query string that the Find adapters service applies to select a subset of the entries that use the specified form. The input signature of the Find adapter service includes this query string, allowing you to pre-define the query string at design time or supply the query string at run time. The output signature of the Find adapter service contains String values that hold the entry ID and description for each matching entry; you cannot modify the output signature.

At run time the Find adapter service uses the AR System Java APIs to locate entries. It locates entries that match the form name and query string you specify. The AR System Server returns the list of matching entries.

Get Adapter Service

Services configured with the Get template obtain the details about an entry in the AR System database. The entry obtained is identified by the name of the AR System form along with the entry ID.

At design time, you identify the AR System form that the Get adapter service is to use. You can also optionally supply the entry ID that the Get Adapter service uses to retrieve the entry details. Alternatively, the entry ID can be supplied at run time. You can configure the output signature

of the Get adapter service to contain the field values that you select. The Get adapter service returns the selected field values from the entry.

At run time, the Get adapter service uses the AR System Java APIs to obtain the selected field values of the specified entry. The AR System Server returns the field values from the entry.

Batch Get Adapter Service

Services configured with the Batch Get template retrieve details about multiple form entries from the Remedy ARS Server. The entries obtained are identified by the name of the AR System form along with the entry IDs.

At design time, you identify the AR System form associated with the Batch Get adapter service and the entry form fields retrieved by this service. As input to the service, you provide the list of entry IDs for the Remedy form that should be retrieved. The Batch Get service returns a document list. The document list takes the name of the Remedy form whose entries are being retrieved. Each document in the list contains the entry data for all entry fields that you selected during design time.

At run time, the Batch Get adapter service uses the AR System Java APIs to obtain the selected field values of the specified entries. The AR System Server returns the field values from the entries.

Update Adapter Service

Services configured with the Update template update fields for an entry in the AR System database. The entry that the Update adapter service updates is identified by the name of the AR System form for the entry along with the entry ID for the entry.

At design time, you identify the AR System form that the Update adapter service is to use. You can also optionally supply the entry ID of the entry that the Update Adapter service uses to update the details. Alternatively, the entry ID can be supplied at run time.

You configure the input signature of the Update adapter service to contain the fields that you select. The output signature of the Update adapter service contains a String that holds the status (or outcome) of the update operation (Success or Failure); you cannot modify the output signature.

At run time, the Update adapter service uses the AR System Java APIs to supply new field values for the selected fields. The adapter service retrieves the new values for the fields from the pipeline. The AR System Server returns the result of the update operation, which indicates the status of the update operation.

ARS Server Operations Adapter Service

Services configured with the ARS Server Operations template execute one of the following ARS Server operations:

Operation	Description
Execute Process	Executes the input command on the AR System server and returns the result if the service has been configured to wait for the process to

Operation	Description
	complete. For more information about the commands that can be executed on the AR System server, see the BMC Remedy Action Request System documentation.
Get Operation Time	Gets the operation time stamp set by the AR System server after each API call.
Begin Bulk Entry Transaction	Marks the beginning of a bulk entry transaction for the associated service connection object.
End Bulk Entry Transaction	Marks the ending of a bulk entry transaction for the associated service connection object.

Note:

The Bulk Entry Transaction operations supported in Remedy Adapter are different than any other WmART-based adapter transactions. The Remedy Adapter Bulk Entry Transaction operations are not handled by the WmART-based (Integration Server) transaction manager.

When you select one of the above operations in the service template, the input and output parameters of the adapter service change based on the operation selected.

Using Adapter Services

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

Task	Use this tool...
1 Create and enable an adapter connection. See “Adapter Connections” on page 41 for details.	Integration Server Administrator
2 Select the appropriate adapter service template and configure the necessary adapter services. Depending on the type of adapter service, you specify the: <ul style="list-style-type: none"> ■ Adapter connection ■ Form name for the entry in the AR System with which the adapter service is to interact ■ Optionally, the entry ID for an entry if you are getting information about an entry or deleting an entry See “Adapter Services” on page 51 for more information about configuring each of the adapter services.	Designer
3 If you plan to use an Integration Server flow service or Java Designer service to invoke the adapter service, design the flow service to use Remedy Adapter services you configure. For	

Task	Use this tool...	
	information about using Designer, see the <i>webMethods Service Development Help</i> for your release.	
4	Manage the adapter services. See “ Adapter Package Management ” on page 31 and Adapter Logging and Exception Handling for details.	Designer and Integration Server Administrator

Using Version Control Systems to Manage Adapter Elements

The adapter supports the Version Control System (VCS) Integration feature provided by Designer. When you enable the feature in Integration Server, you can check adapter packages or elements into and out of your version control system from Designer. For more information about the VCS Integration feature, see the *Configuring the VCS Integration Feature*.

Beginning with Integration Server 8.2 SP3, the adapter supports the local service development feature in Designer. This feature extends the functionality of the VCS Integration feature to check package elements and their supporting files into and out of a VCS directly from Designer. For more information about local service development and how it compares to the VCS Integration feature, see the *webMethods Service Development Help*.

Optimize Infrastructure Data Collector Support for the Adapter

Optimize Infrastructure Data Collector monitors the system and operational data associated with webMethods run-time components such as Integration Servers, Broker Servers, Brokers, and adapters, and reports the status of these components on Optimize for Infrastructure or other external tools. When you start monitoring an Integration Server, Infrastructure Data Collector automatically starts monitoring all ART-based adapters that are installed on the Integration Server.

For information about monitored key performance indicators (KPIs) collected for the monitored adapter components, see the *webMethods Optimize User's Guide* for your release.

Viewing the Adapter's Update Level

With Integration Server 7.1 or later, 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 the Integration Server Administrator.

Controlling Pagination

When using the adapter on Integration Server 8.0 and later, you can control the number of items that are displayed on the adapter Connections screen and Notifications 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.
4. For more information about working with extended configuration settings, see the *webMethods Integration Server Administrator's Guide* for your release.

2 Installing webMethods Remedy Adapter

■ Overview	28
■ Requirements	28
■ Installing Remedy Adapter 7.1	28
■ Upgrading from Remedy Adapter 6.0	29
■ Uninstalling Remedy Adapter 7.1	30

Overview

This chapter explains how to install, upgrade, and uninstall webMethods Remedy Adapter 7.1. The instructions use the Software AG Installer and the 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 the operating systems, AR System products, and webMethods products supported by Remedy Adapter 7.1, see the *webMethods Adapters System Requirements*.

Remedy Adapter 7.1 has no hardware requirements beyond those of its host Integration Server.

Installing Remedy Adapter 7.1

If you are using Remedy Adapter 6.0, see [“Upgrading from Remedy Adapter 6.0” on page 29](#).

Note:

If you are installing the adapter in a clustered environment, you must install it on each Integration Server in the cluster, and each installation must be identical. For more information about working with the adapter in a clustered environment, see [“Adapter Package Management” on page 31](#).

➤ To install Remedy Adapter 7.1

1. If you are installing the adapter on an existing Integration Server, shut down the Integration Server.
2. Download Software AG Installer from the [Empower Product Support Web site](#).
3. Start the Installer wizard.
4. Choose the webMethods release that includes the Integration Server on which to install the adapter. For example, if you want to install the adapter on Integration Server 8.0, choose the 8.x release.
5. Specify the installation directory as follows:
 - If you are installing on an existing Integration Server, specify the Software AG installation directory that contains the host Integration Server.
 - If you are installing both the host Integration Server and the adapter, specify the installation directory to use.

The Installer will install the adapter in the *Integration Server_directory \packages* directory.

6. In the product selection list, select **Adapters > webMethods Remedy Adapter 7.1**. You can also choose to install documentation.

Alternatively, you can download the adapter documentation at a later time from the [Software AG Documentation Web Site](#).

7. After installation is complete, deploy the AR System library files on the host Integration Server as follows:

Note:

Deploy the library files for the operating system on which Integration Server is installed; for example, if the Integration Server is installed on a Solaris system, deploy the library files for Solaris.

- Copy the relevant AR System jar files to the *Integration Server_directory \packages \WmRemedyAdapter \code \jars* directory. For a list of the AR System jar files for each AR System version and the exact source location of the jar files, see the *webMethods Adapters System Requirements*.
- For AR System 7.1 along with the relevant AR System jar files (see the previous bullet), copy the required native libraries to the *Integration Server_directory \lib* directory. For information about the native libraries, see the *Remedy AR System API Reference Guide* for AR System 7.1.

For more information about the Java API requirements and C API package contents of the BMC Remedy AR System 7.1 or later, see the respective *Remedy AR System API Reference Guide* for each version.

8. Start the host Integration Server.

Upgrading from Remedy Adapter 6.0

Remedy Adapter 7.1 is compatible with Remedy Adapter 6.0. You can use Remedy Adapter 7.1 to run or edit the services generated using Remedy Adapter 6.0.

Upgrading from Remedy Adapter 6.0

Important:

Before running the Remedy Adapter `migrate_60_To_71` service, you must enable the adapter connection associated with the service in the target package.

➤ To upgrade from Remedy Adapter 6.0 to Remedy Adapter 7.1

1. Back up your existing Remedy Adapter 6.0 installation and all custom packages.
2. Uninstall Remedy Adapter 6.0 as described in *webMethods Remedy Adapter Installation Guide 6.0*.

3. Delete the dlls and the lib files of Remedy Adapter 6.0. Uninstalling the adapter does not delete those files.
4. Install Remedy Adapter 7.1 as described in [“Installing Remedy Adapter 7.1” on page 28](#).
5. To migrate services created with version 6.0 of Remedy Adapter to version 7.1:
 - a. Enable the adapter connections associated with the target package.
 - b. Run the `wm.adapter.wmremedy.util:migrate_60_To_71` service located in the `WmRemedyAdapter` package. For information about the service, see [“wm.adapter.wmremedy.util:migrate_60_To_71” on page 104](#).

You can run this service using either Designer. After running the `migrate_60_To_71` service, the templates of the adapter services created with Remedy Adapter 6.0 will be updated to the Remedy Adapter 7.1 templates.

Uninstalling Remedy Adapter 7.1

➤ To uninstall Remedy Adapter 7.1

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 Software AG installation directory that contains the host Integration Server. In the product selection list, select **Adapters > webMethods Remedy Adapter 7.1**. You can also choose to uninstall documentation.
3. Restart the host Integration Server.
4. Uninstaller removes all Remedy Adapter 7.1-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 delete the `WmRemedyAdapter` directory.

3 Adapter Package Management

■ Overview	32
■ Managing the Adapter Package	32
■ Controlling Group Access	35
■ Using Remedy Adapter in a Clustered Environment	36

Overview

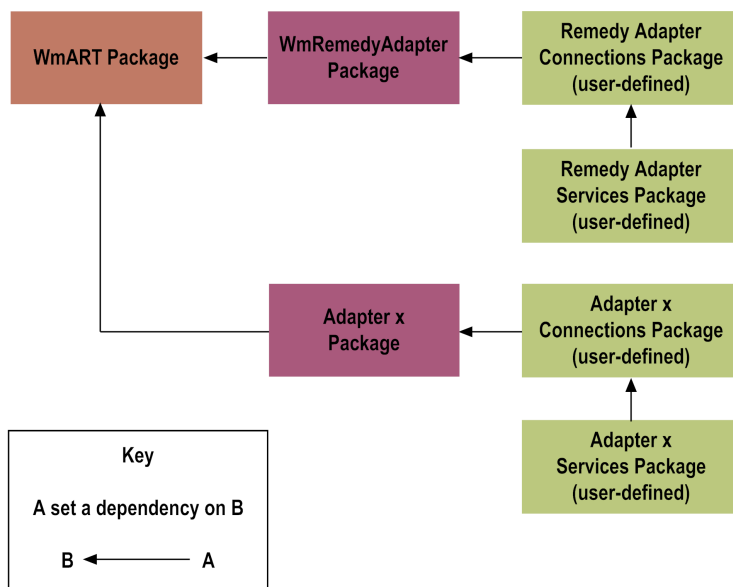
The following sections describe how to set up and manage your Remedy Adapter packages, set up Access Control Lists (ACL), and use the adapter in a clustered environment.

Managing the Adapter Package

Remedy Adapter is provided as a package called WmRemedyAdapter. You manage the WmRemedyAdapter package as you would manage any package on Integration Server.

When you create connections and adapter services, define them in user-defined packages rather than in the WmRemedyAdapter 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 Designer and set the user-defined packages to have a dependency on the WmRemedyAdapter package. That way, when the WmRemedyAdapter 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 33](#)).
- [“Enabling Packages” on page 33](#).
- [“Disabling Packages” on page 34](#).
- [“Controlling Group Access” on page 35](#).

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, WmRemedyAdapter. (The WmRemedyAdapter package has a dependency on the WmART package.)
- Package dependencies ensure that at startup Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and the user-defined packages last. 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 that 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 33](#).
- 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 34](#).
- You can name connections and adapter services the same name provided that they are in different folders and packages.

Enabling Packages

All packages are automatically enabled by default.

➤ To enable a package

1. Open the Integration Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.

- Click **No** in the **Enabled** column. The server displays a ✓ and **Yes** in the **Enabled** column.

Note:

Enabling an adapter package will not cause its associated user-defined packages to be reloaded. For information about reloading packages, see [“Loading, Reloading, and Unloading Packages”](#) on page 34.

Important:

Before you manually enable a user-defined package, you must first enable its associated adapter package (WmRemedyAdapter). Similarly, if your adapter has multiple user-defined packages, and you want to disable some of them, disable the adapter package first. Otherwise, errors will be issued when you try to access the remaining enabled user-defined packages.

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. A disabled package will remain disabled until you explicitly enable it using the Integration Server Administrator.

➤ To disable a package

- Open the Integration Server Administrator if it is not already open.
- In the **Packages** menu of the navigation area, click **Management**.
- 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 package will:
 - Remain disabled until you explicitly enable it using the Integration Server Administrator.
 - Not be listed in Designer. Also its adapter connections will not be available.

Loading, Reloading, and Unloading Packages

Recall that if user-defined packages are properly configured with a dependency on the adapter package (as described in [“Package Dependency Requirements and Guidelines”](#) on page 33), at startup Integration Server automatically loads or reloads all packages in the proper order: the WmART package first, the adapter package next, and the node packages last. You should not need to manually reload the WmART package.

Reloading Packages Manually

Reloading a user-defined package will *not* cause its associated adapter package to be reloaded. You can reload adapter packages and user-defined packages from either the Integration Server Administrator (by clicking the **Reload** icon on the Management window) or from Designer (by right-clicking the package and selecting the **Reload Package** option from the menu).

Unloading Packages

At shutdown, the Integration Server unloads packages in the reverse order in which it loaded them: it unloads the node packages first, the adapter package next, and the WmART package last (assuming the dependencies are correct).

Setting Package Dependencies

You set package dependencies if a given package needs services in another package to load before it can load. For example, any packages you create for Remedy Adapter services should identify the Remedy Adapter package (WmRemedyAdapter) as a package dependency because they require services in the WmRemedyAdapter to load first. Use the following guidelines:

- If you store adapter connections in a different package than adapter services, be sure to:
 - Set the package dependencies of the package containing the adapter connections to depend on the WmRemedyAdapter package, and
 - Set the package dependencies of the adapter services package to depend on the package that contains the adapter connections. That is, the package that contains the connection should load before the adapter service package.

When you set this package dependency, it ensures that if someone disables the connection package and then re-enables it, the adapter services will reload correctly.

- If both the connection and adapter services are in the same package, set the package dependencies of this package to have to depend on the WmRemedyAdapter package.

For more information about setting package dependencies, see the *webMethods Service Development Help* for your release.

Controlling Group Access

To control which development group has access to which adapter services, use access control lists (ACLs). 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 general information about assigning and managing ACLs, see the *webMethods Service Development Help* for your release.

Using Remedy Adapter in a Clustered Environment

What is webMethods Integration Server Clustering?

Clustering is an advanced feature of the webMethods product suite that substantially extends the reliability, availability, and scalability of webMethods Integration Server. Clustering accomplishes this by providing the infrastructure and tools to deploy multiple Integration Servers 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.

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

Integration Server 8.2 SP2 and higher supports the caching and clustering functionality provided by Terracotta. Caching and clustering are configured at the Integration Server level and Remedy Adapter uses the caching mechanism that is enabled on 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.

Configuring the Adapter in a Clustered Environment

When you configure Remedy Adapter to create adapter services, you must:

- Ensure that each Integration Server in the cluster contains an identical set of packages (see [“Replicating Packages to Integration Servers” on page 36](#)).
- Disable the redirection capability for certain predefined administrative services (see [“Disabling the Redirection of Administrative Services” on page 37](#)).

Replicating Packages to Integration Servers

Every Integration Server in the cluster should contain an identical set of packages that you define using Remedy Adapter; that is, you should replicate the Remedy Adapter 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 chapter on managing packages in the *webMethods Integration Server Administrator's Guide* for your release.

Disabling the Redirection of Administrative Services

As mentioned in [“What is webMethods Integration Server Clustering?” on page 36](#), a server that cannot handle a client's service request can automatically redirect the request to another server in the cluster. However, Remedy Adapter uses certain predefined administrative services that you should not allow to be redirected. These services are used internally when you configure the adapter. If you allow these services to be redirected, your configuration specifications might be saved on multiple servers, which is an undesirable result. For example, if you create two Remedy Adapter services, one might be stored on one server, while the other one might be stored on another server. Remember that all adapter services must reside on all Integration Servers in the cluster.

➤ To disable the redirection of administrative services

1. Shut down Integration Server. For the procedure to do this, see the *webMethods Integration Server Administrator's Guide* for your release.
2. Edit the following file:

Integration Server_directory \config\redir.cnf

3. Add the following line to the file:

```
<value name="wm.art">false</value>
```

4. Save the file and restart Integration Server.

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.

The following considerations and requirements apply to Remedy Adapter in a 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 Server in a given For Example...
cluster must have identical...**

Integration Server versions	One Integration Server in the cluster cannot be version 6.5 and another Integration Server in the cluster be version 8.0 - all servers must be the same version, with the same service packs and fixes (updates) applied.
Adapter packages	All adapter packages on one Integration Server should be replicated to all other Integration Servers in the cluster.
Adapter versions	In the cluster, all Remedy Adapters must be the same version, with the same fixes (updates) applied.
Adapter connections	<p>If you configure a connection to the AR System Server, this connection must appear on all servers in the cluster so that any Integration Server in the cluster can handle a given request identically.</p> <p>If you plan to use connection pools in a clustered environment, see “Considerations When Configuring Connections with Connection Pooling Enabled” on page 38.</p>
Adapter services	<p>If you configure a specific 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.</p> <p>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.</p>

See [“Replicating Packages to Integration Servers”](#) on page 36 for information about replicating adapter packages, connections, and adapter services across multiple Integration Servers in a cluster.

Considerations When Installing Remedy Adapter Packages

For each Integration Server in the cluster, use the standard Remedy Adapter installation procedures for each machine, as described in the [“Installing webMethods Remedy Adapter ”](#) on page 27.

Considerations When Configuring Connections with Connection Pooling Enabled

When you configure an adapter 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 AR System Server.

For example, if you have a cluster of two Integration Servers with a connection configured to an AR System Server 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 AR System Server.

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 connections that can be open at one time.

For information about configuring connections for Remedy Adapter, see [“Configuring Adapter Connections” on page 42](#).

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

4 Adapter Connections

■ Overview	42
■ Before Configuring or Managing Adapter Connections	42
■ Configuring Adapter Connections	42
■ Dynamically Changing a Service's Connection at Run Time	45
■ Viewing Adapter Connection Parameters	45
■ Editing Adapter Connections	47
■ Copying Adapter Connections	48
■ Deleting Adapter Connections	48
■ Enabling Adapter Connections	49
■ Disabling Adapter Connections	49

Overview

This chapter describes how to configure and manage Remedy Adapter connections. For more information about how adapter connections work, see [“Adapter Connections” on page 16](#).

Note:

You must have webMethods administrator privileges to access the Remedy Adapter's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.

Before Configuring or Managing Adapter Connections

➤ **To prepare for configuring or managing an adapter connection**

1. Install webMethods Integration Server and Remedy Adapter on the same machine. See [“Installing webMethods Remedy Adapter ” on page 27](#) for details.
2. Make sure you have webMethods administrator privileges so that you can access the Remedy Adapter's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.
3. Start Integration Server and Integration Server Administrator if it is not already running.
4. Using the Integration Server Administrator, make sure the WmRemedyAdapter package is enabled. See [“Enabling Packages” on page 33](#) for instructions.
5. Using Software AG Designer, create a user-defined package to contain the connection, if you have not already done so. See [“Managing the Adapter Package” on page 32](#) for details.
6. Create your Remedy Adapter connections, as described in [“Configuring Adapter Connections” on page 42](#) below.

Configuring Adapter Connections

When you configure Remedy Adapter connections, you specify information that the adapter uses to connect to an AR System Server.

You configure Remedy Adapter connections using the Integration Server Administrator.

➤ **To configure an adapter connection**

1. Start the Integration Server Administrator if it is not already running.
2. In the **Adapters** menu in the Integration Server Administrator's navigation area, click **Remedy Adapter**.

3. On the Connections screen, click **Configure New Connection**.
4. On the Connection Types screen, click **Remedy Connection** for the connection type.
5. On the Configure Connection Type screen, in the **Remedy Adapter** section, provide values for the following parameters:

Parameter	Description/Action
Package	<p>Required. The package in which to create the connection.</p> <p>You must create the package using Designer before you can specify it using this parameter. For general information about creating packages, see the <i>webMethods Service Development Help</i> for your release.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: Create the connection in a user-defined package rather than in the adapter's package. See “Managing the Adapter Package” on page 32 for other important considerations when creating packages for Remedy Adapter.</p> </div>
Folder Name	Required. The folder in which to create the connection.
Connection Name	<p>Required. The name you want to give the connection. Connection names cannot have spaces or use special characters reserved by Integration Server, or 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.</p>

6. In the **Connection Properties** section, provide values for the following parameters:

Parameter	Description/Action
AR Server Host Name or IP address	The host name of the AR System Server.
User	The user name on the AR System Server, used by Integration Server to establish a connection and perform specific operations (as authorized) on the AR System.
Password	The password for the user account defined in the User field.
Retype Password	The password for the user account defined in the User field again.
AR Server Port	The port number that the connection will use to connect to the AR System Server.
Language / Locale	<p>The language code or locale that the AR System uses.</p> <p>The locale uses the format:</p>

Parameter	Description/Action
	<code>language[_territory[.codeset]][@modifier]</code>
	For example, you might specify <code>en_US.IS08859-15@euro</code> . You can also use the simplified format, for example, <code>en_US</code> or <code>en</code> .

7. In the **Connection Management Properties** section, provide values for the following parameters:

Parameter	Description/Action
Enable Connection Pooling	<p>Enables the adapter to use connection pooling. Default: true.</p> <p>See “Connection Pools” on page 16 for more information about connection pooling in the adapter.</p> <p>If you plan to enable connection pooling in a clustered environment, consider the connection pool size. For details, see “Considerations When Configuring Connections with Connection Pooling Enabled” on page 38.</p>
Minimum Pool Size	The minimum number of connection objects that remain in the connection pool at all times. When the adapter creates the pool, it creates this number of connections. Default: 1.
Maximum Pool Size	The maximum number of connection objects that can exist in the connection pool. When the connection pool has reached its maximum number of connections, the adapter will reuse any inactive connections in the pool or, if all connections are active, it will wait for a connection to become available. Default: 10.
Pool Increment Size	If connection pooling is enabled, this parameter specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size. Default: 1.
Block Timeout	If connection pooling is enabled, this parameter 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. Default: 1000.
Expire Timeout	<p>If connection pooling is enabled, this parameter specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. For example, to specify 10 seconds, specify 10000. Enter 0 to specify no timeout. Default: 1000.</p> <p>Note: The adapter will never violate the Minimum Pool Size parameter. These connections remain in the pool regardless of how long they are inactive.</p>

Parameter	Description/Action
Startup Retry Count	If connection pooling is enabled, this parameter specifies the number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails, before issuing an <code>AdapterConnectionException</code> . Default: 0.
Startup Backoff Timeout	If connection pooling is enabled, this parameter specifies the number of seconds to wait between each attempt to initialize the connection pool. Default: 10.

8. Click **Save Connection**.

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

Dynamically Changing a Service's Connection at Run Time

You can run an adapter service using a connection other than the default connection that was associated with the service when the service was created. To override the default, you must code your flow to pass a value through the pipeline into a service's `$connectionName` field.

For example, you have a flow whose primary purpose is to create an entry on the production AR System Server. However, you want the flow to have the capability to create the entry on the test server, with the decision of which AR System Server to update to be made programmatically at run time. The output signature of the flow's first service contains a field called *Target*, which indicates the target server. The flow could branch based on the value in *Target*:

- If *Target* contains the value `Production`, the second service in the flow, a `Create` adapter service, would ignore `$connectionName`, and as a result, use its default connection to create the entry on the production server.
- However, if *Target* contains the value `Test`, the second service in the flow would use the value in the `$connectionName` from the pipeline and use that connection to update the test server.

Keep in mind these restrictions when using dynamic connections:

- The entry with which an adapter service interacts must be deployed on both the production and test AR System Servers.
- The `$connectionName` field is present only in services Created with Designer.

For more information, see [“Changing the Connection Associated with an Adapter Service at Run Time” on page 17](#).

Viewing Adapter Connection Parameters

You can view a connection's parameters from the Integration Server Administrator or from Designer.

Viewing Adapter Connection Parameters Using Integration Server Administrator


➤ To view the parameters for an adapter connection using the Integration Server Administrator

1. Start the Integration Server Administrator if it is not already running.

When using the adapter with Integration Server 8.0 and later, you can sort and filter the list of connections that appears on the Connections screen.

- To sort information on the Connections screen, click the **Up** and **Down** arrows at the top of the column you want to sort.
- To filter the list of connections:
 1. On the Connections screen, click **Filter Connections**.
 2. Type the criterion by which you want to filter into the **Filter criteria** box. Filtering is based on the node name, not the connection alias. To locate all connections containing specific alphanumeric characters, use asterisks (*) as wildcards. For example, if you want to display all connections containing the string "abc", type *abc* in the **Filter criteria** box.
 3. Click **Submit**. The Connections screen displays the connections that match the filter criteria.
 4. To re-display all connections, click **Show All Connections**.

The Connections screen appears, listing all the current connections. You can control the number of connections that are displayed on this screen. For more information, see *webMethods Integration Server Administrator's Guide*.

2. Make sure the connection is enabled. See [“Enabling Adapter Connections” on page 49](#) for details.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **Remedy Adapter**.
4. On the Connections screen, click the  icon for the connection you want to see.

The View Connection screen displays the parameters for the connection. For descriptions of the connection parameters, see the table of parameters in [“Configuring Adapter Connections” on page 42](#).

5. Click **Return to Remedy Adapter Connections** to return to the Connections screen.

Viewing Adapter Connection Parameters Using Designer

➤ To view the parameters for an adapter connection using Designer.


1. Make sure the connection is enabled. See [“Enabling Adapter Connections” on page 49](#) for details.
2. From the Designer navigation area, open the package and folder in which the connection is located.
3. Double-click the connection you want to view.

The parameters for the connection appear in the **Connection Information** tab. For descriptions of the connection parameters, see the table of parameters in [“Configuring Adapter Connections” on page 42](#).

Editing Adapter Connections

If you want to redefine parameters that a connection uses when connecting to an application server, you can update a connection's parameters using the Integration Server Administrator.

➤ To edit an adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **Remedy Adapter**.
3. Make sure the connection is disabled. See [“Disabling Adapter Connections” on page 49](#) for instructions.
4. On the Connections screen, click the  icon for the connection you want to edit.

The Edit Connection screen displays the current parameters for the connection. Update the connection's parameters by typing or selecting the values you want to specify.


For descriptions of the connection parameters, see the table of parameters in [“Configuring Adapter Connections” on page 42](#).

5. Click **Save Changes** to save the connection and return to the Connections screen.
6. Enable the connection. See [“Enabling Adapter Connections” on page 49](#) for instructions.

Copying Adapter Connections

You can copy an existing Remedy Adapter connection to create a new connection with the same or similar connection properties without retyping all properties for the new connection.

➤ To copy an adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **Remedy Adapter**.
3. Make sure the connection is enabled. See [“Enabling Adapter Connections” on page 49](#) for details.
4. On the Connections screen, click the  icon for the connection you want to copy.

The Copy Connection screen displays the current parameters for the connection you want to copy. Name the new connection and edit any connection parameters as needed by typing or selecting the values you want to specify.

For descriptions of the connection parameters, see the table of parameters in [“Configuring Adapter Connections” on page 42](#).

5. Click **Save Connection Copy** to save the connection and return to the Connections screen.

Deleting Adapter Connections

If you no longer want to use a Remedy Adapter connection, use the following instructions to delete the connection.

If you delete a Remedy Adapter connection, the adapter services that are defined to use the connection will no longer work. If you delete a Remedy Adapter connection, you can assign a different connection to an adapter service and reuse the service. To do this, you use the built-in webMethods function `setAdapterServiceNodeConnection`. For more information, see [“Changing the Connection Associated with an Adapter Service at Design Time” on page 17](#).

➤ To delete an adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **Remedy Adapter**.
3. Disable the connection. See [“Disabling Adapter Connections” on page 49](#) for details.

4. On the Connections screen, click the  icon for the connection you want to delete.

Enabling Adapter Connections


Adapter connections must be enabled before you can create adapter services for those connections.

Note:

When you reload a package that contains enabled connections, the connections will automatically be enabled when the package reloads. If the package contains connections that are disabled, they will remain disabled when the package reloads.

> To enable an adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmRemedyAdapter package is enabled. See [“Enabling Packages” on page 33](#) for details.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **Remedy Adapter**.
4. On the Connections screen, click **No** in the **Enabled** column for the connection you want to enable.

Remedy Adapter enables the adapter connection and displays  and **Yes** in the **Enabled** column.

Disabling Adapter Connections

Adapter connections must be disabled before you can edit or delete the connections.

> To disable an adapter connection

1. Start the Integration Server Administrator if it is not already running.
2. Make sure the WmRemedyAdapter package is enabled. See [“Enabling Packages” on page 33](#) for details.
3. In the **Adapters** menu in the navigation area of the Integration Server Administrator, click **Remedy Adapter**.
4. On the Connections screen, click **Yes** in the **Enabled** column for the connection you want to disable.

Remedy Adapter disables the adapter connection and displays **No** in the **Enabled** column.

5 Adapter Services

■ Overview	52
■ Before Configuring or Managing Adapter Services	52
■ Configuring Create Services	53
■ Configuring Delete Services	55
■ Configuring Find Services	56
■ Configuring Get Services	58
■ Configuring Batch Get Services	60
■ Configuring Update Services	62
■ Configuring ARS Server Operations Services	64
■ Testing Adapter Services	67
■ Viewing Adapter Services	67
■ Editing Adapter Services	67
■ Deleting Adapter Services	68
■ Validating Adapter Service Values	69
■ Reloading Adapter Values	70

Overview

The following sections describe how to configure adapter services that you use to access the Remedy Action Request System.

You can configure the following types of services for use with Remedy Adapter:

- Create; see [“Configuring Create Services” on page 53](#).
- Delete; see [“Configuring Delete Services” on page 55](#).
- Find; see [“Configuring Find Services” on page 56](#).
- Get; see [“Configuring Get Services” on page 58](#).
- Batch Get; see [“Configuring Batch Get Services” on page 60](#).
- Update; see [“Configuring Update Services” on page 62](#).
- ARS Server Operations; see [“Configuring ARS Server Operations Services” on page 64](#).

For a description of the adapter services, see [“Adapter Services” on page 51](#).

Before Configuring or Managing Adapter Services

➤ To prepare to configure or manage an adapter service

1. Start your webMethods Integration Server and the Integration Server Administrator if they are not already running.
2. Make sure you have webMethods administrator privileges so that you can access the Remedy Adapter's administrative screens. For information about setting user privileges, see the *webMethods Integration Server Administrator's Guide* for your release.
3. Using the Integration Server Administrator, make sure the WmRemedyAdapter package is enabled. See [“Enabling Packages” on page 33](#) for instructions.
4. Using the Integration Server Administrator:
 - a. Configure the adapter connection you plan to use with the adapter service. See [“Configuring Adapter Connections” on page 42](#) for instructions.
 - b. Make sure the connection you plan to use with the adapter service is enabled. See [“Enabling Adapter Connections” on page 49](#) for instructions.
5. Start Software AG Designer if it is not already running.

Note:

If you are using Designer, use the Service Development perspective. For more information, see the *webMethods Service Development Help* for your release.

6. Using Designer, create a user-defined package to contain the service 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 WmRemedyAdapter package. For more information about managing packages for the adapter, see [“Managing the Adapter Package” on page 32](#).

Configuring Create Services

A Create adapter service creates an entry in the AR System database based on a selected AR System form. You specify the fields and values to use for the new entry as input to the Create adapter service.

If the Create adapter service successfully creates the new entry, the adapter service returns the *EntryID* output variable, which contains the new entry's ID. If the Create adapter service encountered an error and the entry could not be created, an *ARException* is thrown, and Remedy Adapter logs details about the failure to the Integration Server Server log.

For more information about the template used to create these services, see [“Adapter Service Operation” on page 19](#). For more information about adapter services, see [“Using Adapter Services” on page 23](#).

➤ To configure a Create adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 52](#).
2. Start Designer.
3. If you are using Designer, perform the following:
 - a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **Remedy Adapter** as the adapter type and click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.
6. From the list of available templates, select the **Create** template and do the following:

Select a package and folder to contain the service, type a unique name for the service.
7. Click **Finish**.


The service is created and its parameters and controls are displayed in Designer.

8. In the editor, select the **Schema** tab and specify the following value:

Parameter	Description/Action
Form Name	Select the name of the form that the Create adapter service is to use to add an entry. Remedy Adapter populates the Form Name field with a list of forms that are currently defined in the AR System and that are accessible by the user that is defined for the connection. For more information, see “Configuring Adapter Connections” on page 42 .

Note:

When you select a form in the Form Name field, Remedy Adapter updates fields on the **Input Fields**, **Output Fields**, and **Input/Output** tabs based on the selected form name.

9. Specify the fields to be used to create the new entry:
- Select the **Input Fields** tab. This tab lists the fields that are defined for the AR System form that you selected on the **Schema** tab.
 - For each field that must be supplied when this Create adapter service is invoked, select the check box in the **Use Field** column. If you want to select all fields, click the  icon.

Note:

The information on the **InputFields** tab represents the input that will be required for the Create adapter service.

10. You can select the **Output Fields** tab to view the fields that the Create adapter service returns as output. The output for a Create adapter service is always *EntryID*, which is the entry ID of the newly created entry in the AR System.
11. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter name, connection name, and service template, as needed.
12. The **Input/Output** tab lists the input and output parameters for the Create adapter service. If this service is later used in a flow, its input and output signatures will be visible in the flow editor.

For additional information about using the **Input/Output** tab, see the *webMethods Service Development Help* for your release.

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

Configuring Delete Services

A Delete adapter service deletes an entry from the AR System database. You identify the entry to delete by specifying the AR System form name and entry ID.

If the delete is successful, the Delete adapter service returns the *Status* output variable set to SUCCESS. If the Delete adapter service encountered an error and the entry could not be deleted, an exception is thrown.

For more information about the template used to create these services, see [“Delete Adapter Service” on page 21](#). For more information about adapter services, see [“Using Adapter Services” on page 23](#).

➤ To configure a Delete adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 52](#).
2. Start Designer.
3. If you are using Designer, perform the following:
 - a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **Remedy Adapter** as the adapter type and click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.
6. From the list of available templates, select the **Delete** template and do the following:

Select a package and folder to contain the service, type a unique name for the service.
7. Click **Finish**.

The service is created and its parameters and controls are displayed in Designer.
8. In the editor, select the **Schema** tab and specify the following values:

Parameter	Description/Action
Form Name	Select the name of the form from where the entry is to be deleted.

Parameter	Description/Action
	<p>Remedy Adapter populates the Form Name field with a list of forms that are currently defined in the AR System and that are accessible by the user that is defined for the connection. For more information, see “Configuring Adapter Connections” on page 42.</p> <p>You can configure this form name at design time or supply it at run time. To supply the form name at run time, select the blank item from the list. If you select the blank item, the Delete adapter service will have a <i>formName</i> input variable.</p>
Entry ID	<p>Specify the entry ID of the entry that you want the Delete adapter service to delete.</p> <p>Remedy Adapter populates the Entry ID field with a list of all entry IDs in the AR System that use the selected form.</p> <p>Rather than specifying an entry ID at design time (when you configure the Delete adapter service), you can pass the entry ID to the Delete adapter service as an input at run time.</p> <div>Note: If you want to specify an entry ID at runtime, leave the parameter field blank at design time.</div>

- You can select the **Output Fields** tab to view the fields that the Delete adapter service returns as output. The output for a Delete adapter service is always *Status*, which is the outcome from the delete operation. The output variable, *Status*, is set to SUCCESS when the row is deleted successfully. Otherwise, *Status*, is set to FAILURE, and an exception is thrown.
- You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter name, connection name, and service template, as needed.
- The **Input/Output** tab lists the input and output parameters for the service. If the service is subsequently used in a flow, its input and output signatures will be visible in the flow editor.

If you selected the blank item for **Form Name**, the service will have the input variables *formName* and *EntryID*.

For additional information about using the **Input/Output** tab, see the *webMethods Service Development Help* for your release.

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

Configuring Find Services

A Find adapter service locates one or more entries that are stored in the AR System database. You identify the entries you want to find by specifying the name of the AR System form used for the

entries and providing a query string to use as search criteria. The Find adapter service finds entries for the specified form that match the query string.

Note:

The Find adapter service returns the entry IDs of matching entries. It does not return the details about entries. If you want to access the details, for example, the values of the fields associated with an entry, use the Get adapter service. See [“Configuring Get Services” on page 58](#).

For more information about the template used to create these services, see [“Find Adapter Service” on page 21](#). For more information about adapter services, see [“Using Adapter Services” on page 23](#).

➤ **To configure a Find adapter service**

1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 52](#).
2. Start Designer.
3. If you are using Designer, perform the following:
 - a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **Remedy Adapter** as the adapter type and click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.
6. From the list of available templates, select the **Find** template and do the following:
Select a package and folder to contain the service, type a unique name for the service.
7. Click **Finish**.
The service is created and its parameters and controls are displayed in Designer.
8. In the editor, select the **Query** tab and specify the following values:

Parameter	Description/Action
Form Name	<p>Select the name of the form that was used for the entries that you want to find.</p> <p>Remedy Adapter populates the Form Name field with a list of forms that are currently defined in the AR System and that are accessible by</p>

Parameter	Description/Action
	<p>the user that is defined for the connection. For more information, see “Configuring Adapter Connections” on page 42.</p> <p>You can configure this form name at design time or supply it at run time. To supply the form name at run time, select the blank item from the list. If you select the blank item, the Find adapter service will have a <i>formName</i> input variable.</p>
Query String	<p>Type a query string to narrow down the list of entries, for example, ‘Request ID’ > 10. The query string will be applied against all entries that use the selected form. The Find adapter service returns the matching entries.</p> <p>Rather than specifying a query string at design time when you configure the Find adapter service, you can pass the query string to the Find adapter service as input at run time, for example, <code>\${runtime_var}</code>.</p> <p>If you do not specify a query string, the Find adapter service returns all entries that use the form you selected in the Form Name field.</p>

9. You can select the **Output Fields** tab to view the fields that the Find adapter service returns for each matching entry. These fields are:
 - *EntryId*, which is the entry ID for the matching entry
 - *Description*, which is a description for the entry.
10. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter name, connection name, and service template, as needed.
11. The **Input/Output** tab lists the input and output parameters for the service. If the service is subsequently used in a flow, its input and output signatures will be visible in the flow editor.

If you selected the blank item for **Form Name**, the service will have the input variables *formName*.

For additional information about using the **Input/Output** tab, see the *webMethods Service Development Help* for your release.

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

Configuring Get Services

A Get adapter service retrieves the details for an existing entry in the AR System database. You identify the entry by specifying the name of the AR System form used for the entry along with the entry ID for the entry.

When retrieving field values for an entry, the Remedy Adapter maps the data type of the field from an AR System data type to a data type that the Remedy Adapter uses. The Remedy Adapter does not support all AR System data types. The Remedy Adapter attempts to use `java.lang.String` for unsupported data types. For more information, see [“Data Mapping” on page 99](#).


For more information about the template used to create Get adapter services, see [“Get Adapter Service” on page 21](#). For more information about adapter services, see [“Using Adapter Services” on page 23](#).

» To configure a Get adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 52](#).
2. Start Designer.
3. If you are using Designer, perform the following:
 - a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **Remedy Adapter** as the adapter type and click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.
6. From the list of available templates, select the **Get** template and using Designer, click **Finish**.
The service is created and its parameters and controls are displayed in Designer.
7. In the editor, select the **Schema** tab and specify the following values:

Parameter	Description/Action
Form Name	<p>Select the name of the form that was used for the entry.</p> <p>Remedy Adapter populates the Form Name field with a list of forms that are currently defined in the AR System and that are accessible by the user that is defined for the connection. For more information, see “Configuring Adapter Connections” on page 42.</p>
Entry ID	<p>Specify the entry ID of the entry that you want the Get adapter service to retrieve.</p> <p>Remedy Adapter populates the Entry ID field with a list of all entry IDs in the AR System for the selected form.</p>

Parameter	Description/Action
	Rather than specifying an entry ID at design time when you configure the Get adapter service, you can pass the entry ID to the Get adapter service as input at run time.

8. To specify the fields that you want the Get adapter service to retrieve for the entry:
 - a. Select the **Output Fields** tab. This tab lists the fields that are defined for the AR System form that you selected on the **Schema** tab.
 - b. For each field that you want the Get adapter service to retrieve, select the fields by selecting the check box in the **Use Field** column. If you want to select all fields, click the  icon.
9. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter name, connection name, and service template, as needed.
10. The **Input/Output** tab lists the input and output parameters for the adapter service. If this service is later used in a flow, its input and output signatures will be visible in the flow editor.

For additional information about using the **Input/Output** tab, see the *webMethods Service Development Help* for your release.

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

Configuring Batch Get Services

A Batch Get adapter service retrieves the details about multiple form entries from the Remedy ARS Server. The entries obtained are identified by the name of the AR System form along with the entry IDs.

When retrieving field values for an entry, Remedy Adapter maps the data type of the field from an AR System data type to a data type that Remedy Adapter uses. Remedy Adapter does not support all AR System data types. Remedy Adapter attempts to use `java.lang.String` for unsupported data types. For more information, see [“Data Mapping” on page 99](#).

For more information about the template used to create Batch Get adapter services, see [“Batch Get Adapter Service” on page 22](#). For more information about adapter services, see [“Using Adapter Services” on page 23](#).

➤ To configure a Batch Get adapter service


1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 52](#).
2. Start Designer.
3. If you are using Designer, perform the following:

- a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **Remedy Adapter** as the adapter type and click **Next**.
 5. Select the appropriate **Adapter Connection Name** and click **Next**.
 6. From the list of available templates, select the **Batch Get** template and do the following:
Select a package and folder to contain the service, type a unique name for the service.
 7. Click **Finish**.

The service is created and its parameters and controls are displayed in Designer.

8. In the editor, select the **Schema** tab and specify the following values:

Parameter	Description/Action
Form Name	<p>Select the name of the form that was used for the entry.</p> <p>Remedy Adapter populates the Form Name field with a list of forms that are currently defined in the AR System and that are accessible by the user that is defined for the connection. For more information, see “Configuring Adapter Connections” on page 42.</p>
Entry IDs	<p>Specify the entry IDs of the entries that you want the Batch Get adapter service to retrieve.</p> <p>Remedy Adapter populates the Entry IDs field with a list of all entry IDs in the AR System for the selected form.</p> <p>Rather than specifying the entry IDs at design time when you configure the Batch Get adapter service, you can pass the entry IDs to the Batch Get adapter service as input at run time.</p>

9. To specify the fields that you want the Batch Get adapter service to retrieve for the entry:
 - a. Select the **Output Fields** tab. This tab lists the fields that are defined for the AR System form that you selected on the **Schema** tab.
 - b. For each field that you want the Batch Get adapter service to retrieve, select the fields by selecting the check box in the **Use Field** column. If you want to select all fields, click the  icon.

10. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter name, connection name, and service template, as needed.
11. The **Input/Output** tab lists the input and output parameters for the adapter service. If this service is later used in a flow, its input and output signatures will be visible in the flow editor.

For additional information about using the **Input/Output** tab, see the *webMethods Service Development Help* for your release.

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

Configuring Update Services

An Update adapter service updates fields for an existing entry in the AR System database. You identify the entry to update by specifying the name of the AR System form along with the entry ID for the entry. At run time, the values of the fields to be updated are passed as input. For more information about the template used to create these services, see [“Update Adapter Service” on page 22](#). For more information about adapter services, see [“Using Adapter Services” on page 23](#).

➤ To configure an Update service


1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 52](#).
2. Start Designer.
3. If you are using Designer, perform the following:
 - a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **Remedy Adapter** as the adapter type and click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.
6. From the list of available templates, select the **Update** template and do the following:

Select a package and folder to contain the service, type a unique name for the service.
7. Click **Finish**.

The service is created and its parameters and controls are displayed in Designer.

8. In the editor, select the **Schema** tab and specify the following values:

Parameter	Description/Action
Form Name	<p>Select the name of the form that was used for the entry that you want to update.</p> <p>Remedy Adapter populates the Form Name field with a list of forms that are currently defined in the AR System and that are accessible by the user that is defined for the connection. For more information, see “Configuring Adapter Connections” on page 42.</p>
Entry ID	<p>Specify the entry ID of the entry that you want to update.</p> <p>Remedy Adapter populates the Entry ID field with a list of all entry IDs in the AR System that use the selected form.</p> <p>Rather than specifying an entry ID at design time when you configure the Update adapter service, you can pass the entry ID to the Update adapter service as input at run time.</p>

9. Specify the fields that you want the Update adapter service to update:
- Select the **Input Fields** tab. This tab lists the fields that are defined for the AR System form that you selected on the **Schema** tab.
 - For each field that you want the Update adapter service to update, select the fields by selecting the check box in the **Use Field** column. If you want to select all fields, click  icon.
10. You can select the **Output Fields** tab to view the fields that the Update adapter service returns as output. The output for a Update adapter service is always *Status*, which is the outcome from the update operation (Success or Failure).
11. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter name, connection name, and service template, as needed.
12. The **Input/Output** tab lists the input and output parameters for the service. If the service is subsequently used in a flow, its input and output signatures will be visible in the flow editor.
- For additional information about using the **Input/Output** tab, see the *webMethods Service Development Help* for your release.
13. From the **File** menu, select **Save**.

Configuring ARS Server Operations Services

An ARS Server Operations service executes one of the supported ARS Server operations. When you select one of the supported server operations in the service template, the input and output parameters of the adapter service change based on the operation selected.

If the server operation is successful, the adapter service returns the *status* output variable set to SUCCESS and the following output for each operation:

Operation	Output on SUCCESS status
Execute Process	<p>The output of the process:</p> <ul style="list-style-type: none">■ <i>statusCode</i> The AR System Java client API returns an integer identifying the status of the operation. A value of 0 indicates success. Any other value indicates a failure.■ <i>statusMessage</i> The AR System Java client API returns a string containing the process output. Based on the outcome of the operation, this string contains either result data or an error message. <p>For more information about the AR System JAVA client API, see the BMC Remedy documentation.</p>
Get Operation Time	<p>The timestamp of the operation.</p> <p><i>statusMessage</i> The AR System Java client API returns a string containing the process output. Based on the outcome of the operation, this string contains either result data or an error message. For more information about the AR System JAVA client API, see the BMC Remedy documentation.</p>
Begin Bulk Entry Transaction	<p>Marks the beginning of a Bulk Entry Transaction:</p> <p><i>statusMessage</i> The value is a status message that the AR System server returns <i>only</i> when a failure occurs.</p>
End Bulk Entry Transaction	<p>If the execution of a queued call resulted in a new entry as output, the new entry ID:</p> <ul style="list-style-type: none">■ <i>results</i> Array of IData (IS Document type). Each result document contains the following fields:<ul style="list-style-type: none">■ <i>callType</i> An integer that indicates the operation type. For example, 1 for create operation, 2 for update operation.■ <i>entryID</i> The entry ID value of the new entry. This value is returned only for the create operation. In all other cases the value is null.■ <i>statusMessage</i> The value is the status message for the associated Bulk Entry Transaction operation that the AR System server returns.

Operation	Output on SUCCESS status
	For information about the Bulk Entry Transaction results, see the BMC Remedy documentation.

If the ARS Server Operations adapter service encountered an error and the operation could not be executed, the adapter service returns the *status* output variable set to FAILURE and Remedy Adapter logs details about the failure into the Integration Server server log.

For more information about the template used to create these services, see [“ARS Server Operations Adapter Service” on page 22](#). For more information about adapter services, see [“Using Adapter Services” on page 23](#).

➤ To configure an ARS Server Operations service

1. Review the steps in [“Before Configuring or Managing Adapter Services” on page 52](#).
2. Start Designer.
3. Using Designer, perform the following:
 - a. Right-click the package in which the service should be contained and select **New > Adapter Service**.
 - b. Select the parent namespace and type a name for the adapter service.
 - c. Click **Next**.
4. Select **Remedy Adapter** as the adapter type and click **Next**.
5. Select the appropriate **Adapter Connection Name** and click **Next**.

Important:

You must use the same adapter connection for the Begin Bulk Entry Transaction and the End Bulk Entry Transaction operations. Only the Begin Bulk Entry Transaction and End Bulk Entry Transaction adapter services executed with the same adapter connection will be part of a transaction. Begin and End Bulk Entry Transaction service pairs that use the same adapter connection cannot be nested within one another.

6. From the list of available templates, select the **ARS Server Operations** template and do the following:

Select a package and folder to contain the service, type a unique name for the service.

7. Click **Finish**.

The service is created and its parameters and controls are displayed in Designer.

8. In the editor, in the **Operation** field on the **ARS Server Operation** tab, select the ARS Server operation that you require.

When you select one of the ARS Server operations, the input and output parameters of the adapter service change based on the operation selected.

9. Specify the following values based on the selected ARS Server operation:

Operation	Parameter	Description/Action
Execute Process	<i>command</i>	Specify the command to execute on the AR System Server.
	<i>wait</i>	Specify whether the service waits for the process to complete. Valid values: <ul style="list-style-type: none">■ TRUE The service waits for the process to complete.■ FALSE The service does not wait for the process to complete.
Get Operation Time	None.	
Begin Bulk Entry Transaction	None.	
End Bulk Entry Transaction	<i>send</i>	Specify whether the adapter transmits the queued entry calls to the server. Valid values:
		<ul style="list-style-type: none">■ TRUE The service transmits the queued entry calls to the server and ends the transaction.■ FALSE The service removes the queued entry calls and ends the transaction.

10. You can select the **Adapter Settings** tab at any time to confirm adapter properties such as adapter name, connection name, and service template, as needed.
11. The **Input/Output** tab lists the input and output parameters for the adapter service. If this service is later used in a flow, its input and output signatures will be visible in the flow editor.

For additional information about using the **Input/Output** tab, see the *webMethods Service Development Help* for your release.

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

Testing Adapter Services

You use Designer to test adapter services.

For more information about testing and debugging services, see the *webMethods Service Development Help* for your release.

➤ To test adapter services

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 52.
2. In Designer, expand the package and folder that contain the service you want to test.
3. Double-click the service you want to test.

Designer displays the configured service in the service template's Adapter Service Editor.

4. Using Designer, select **Run > Run As > Run Service**.
5. For every service input field, you will be prompted to enter an input value. Enter a value for each input field and then click **OK**.
6. Click the **Service Result** tab (in Designer) to view the output from this service.

Viewing Adapter Services

You use Designer to view adapter services.

➤ To view an adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 52.
2. In Designer, expand the package and folder that contain the service you want to view.
3. Double-click the service you want to view.

Designer displays the configured service in the service template's Adapter Service Editor.

Editing Adapter Services

You use Designer to edit adapter services.

➤ To edit an adapter service

1. In Designer, browse to and open the adapter service that you want to edit.

2. Double-click the service that you want to edit.

Designer displays the adapter service in the service template's Adapter Service Editor.

3. Do one of the following:

- If you have the VCS Integration feature enabled, right-click the service and select **Check Out**.
- If you do not have the VCS Integration feature enabled, right-click the service and select **Lock for Edit**.
- If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check out the service. The options available in the **Team** menu depend on the VCS client that you use.

4. Modify the values for the adapter service's parameters as needed. For detailed descriptions of the service's parameters, see the section on configuring a service for the specific type of service you want to edit.

5. After you complete your modifications, save the service and do one of the following:

- If you have the VCS Integration feature enabled, right-click the service and select **Check In**. Enter a check-in comment and click **OK**.
- If you do not have the VCS Integration feature enabled, right-click the service and select **Unlock**.
- If you are using the local service development feature, from the **Team** menu in Designer, select the appropriate option to check in the service. The options available in the **Team** menu depend on the VCS client that you use.

6. Save the service.

Deleting Adapter Services

You use Designer to delete adapter services.

➤ To delete an adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 52.
2. In Designer, expand the package and folder that contain the service you want to delete.
3. Right-click the service and click **Delete**.

Validating Adapter Service Values

Designer enables Remedy Adapter to validate user-defined data for adapter services at design time. You can validate the values for a single adapter service or you can configure Designer to always validate the values for adapter services. Both options could potentially slow your design-time operations.

When you enable data validation for a single adapter service, Designer compares the service values against the resource data that has already been fetched from the selected adapter.

If you select the option to always validate values for adapter services, it will do so for all webMethods WmART-based adapters installed on Integration Server.


For more information about the **Adapter Service/Notification Editor** and other Designer menu options and toolbar icons, see the *webMethods Service Development Help* for your release.

Validating for a Single Adapter Service

➤ To enable automatic data validation for a single adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 52.
2. In Designer, expand the package and folder that contain the service for which you want to enable automatic validation.
3. Double-click the service for which you want to validate the data.

Designer displays the configured adapter service in the service template's Adapter Service Editor.

4. Click the  icon.

Validating for All Adapter Services

➤ To always validate the values for all adapter services

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 52.
2. Start Designer.
3. using Designer, select the **Window > Preferences > Software AG > Service Development > Adapter Service/Notification Editor** item.
4. Enable the **Automatic data validation** option.

5. Click **OK**.

Reloading Adapter Values

You can enable the Remydy Adapter to reload and validate user-defined data for adapter services at design time in Designer. You can reload values for a single adapter service or you can configure Designer so it automatically reloads the values for adapter services. Both options could potentially slow your design-time operations.

When you reload adapter values for a single adapter service, Designer compares the service values against the resource data that has already been fetched from the selected adapter.

If you select the option to always reload values for adapter services, it will do so for all webMethods WmART-based adapters installed on the Integration Server.


For more information about the **Adapter Service/Notification Editor**, other menu options, and toolbar icons, see the *webMethods Service Development Help* for your release.

Reloading for a Single Adapter Service

➤ To reload the adapter values for a single adapter service

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 52.
2. In Designer, expand the package and folder that contain the service for which you want to enable automatic validation.
3. Double-click the service for which you want to validate the data.

Designer displays the configured adapter service in the service template's Adapter Service Editor.

4. Click the  icon.

Reloading for All Adapter Services

➤ To reload the adapter values for all adapter services

1. Review the steps in [“Before Configuring or Managing Adapter Services”](#) on page 52.
2. Start Designer.
3. Using Designer, select the **Window > Preferences > Software AG > Service Development > Adapter Service/Notification Editor** item.

4. Enable the **Automatic polling of adapter metadata** option.
5. Click **OK**.

6 Configuring Remedy to webMethods Notifications

■ Overview	74
■ Setting Up E-mail Notifications	74
■ Setting Up Web Services	78

Overview

You can set up filters in the AR System to detect when changes occur and to send webMethods Integration Server notification of the changes. You set up Integration Server to process the documents that the AR System sends. Based on the changes on the AR System, you can take appropriate actions in Integration Server. For an overview of this processing, see [“Notifications Sent from the AR Systems” on page 13](#).

When you set up the filter in the AR System, you define the action in the filter to either:

- **Send information via an e-mail message.** For information about how to set up the AR System to send this type of notification, see the Remedy AR System documentation. For information about how to set up Integration Server to process the notification, see [“Setting Up E-mail Notifications” on page 74](#).
- **Invoke a Web service to which the AR System passes information in an XML document.** For information about how to set up the AR System to send this type of notification, see the Remedy AR System documentation. For information about how to set up Integration Server to process the notification, see [“Setting Up Web Services” on page 78](#).

Setting Up E-mail Notifications

You can set up the AR System to send an e-mail message to notify Integration Server when a change is made to entries that the AR System maintains. The AR System can send e-mail messages of the following types:

- Formatted text
- HTML templates
- MIME attachments
- Customizable e-mail header fields

You create a service on Integration Server to process the e-mail notification. Your service can take an appropriate action based on the change to an entry that the AR System maintains.

For an overview of how the AR System sends an e-mail message to Integration Server for handling, see [“Receiving an E-mail Notification” on page 13](#).

The following table lists the actions you must take in Integration Server to set up sending e-mail notifications and processing them.

Step	Tool	Action
1	Designer or Digital Event Services	Create a service that receives the e-mail from the AR System and takes appropriate actions.
2	Integration Server Administrator	Configure an Email Listener port that will listen for incoming e-mail messages on the specified Mail server.

Creating a Service to Handle the Notification

When the Email Listener, which you create in the next step, receives the e-mail message from the AR System, it invokes a service to handle the e-mail message. You create this service to handle and process the received e-mail message.

You create the service to receive the e-mail message that the AR System sends and process the message as you require. Remedy Adapter provides the sample `wm.adapter.wmremedy.outbound.emailNotification:receiveEmail` service to illustrate how to receive and process an e-mail message. This sample service receives an e-mail message and invokes another sample service, `wm.adapter.wmremedy.outbound.emailNotification:readEmailContent`, which is a Java service. The `readEmailContent` service processes the content of the e-mail message by extracting the body of the single e-mail message, as well as the message with an attachment. You can modify this service for your own use.

Configuring an Email Listener Port in Integration Server

From the Integration Server Administrator, you configure an Email Listener port that listens for e-mail messages that the AR System sends to a specified mail server. When the e-mail listener detects a new e-mail message, it extracts this message. Integration Server then invokes a service that you define to handle the e-mail message.

For detailed information about creating an e-mail listener in Integration Server, see the *webMethods Integration Server Administrator's Guide* for your release.

➤ To configure an Email Listener Port in Integration Server

1. Open the Integration Server Administrator.
2. From the **Security** menu of the navigation panel, select **Ports**.
3. Click **Add Port**.
4. Select **webMethods/Email**, and click **Go to Step 2**.
5. In the **Package** section of the screen, select a package in which to create the Email Listener port.

You must create the package using Designer or Digital Event Services before you can specify it using this parameter. For general information about creating packages, see the *webMethods Service Development Help* for your release.

6. Fill in the following fields in the **Server Information** section of the screen to identify the mail server to check for incoming e-mail messages.

In this field...	Specify...
Host Name	Name of the machine on which the mail server is running (for example, mailserver.company.com). This must be the same mail server you identified when creating the outgoing mailbox in the AR System.
Type	Type of mail server. Select POP3 or IMAP .
User Name	User name of the user to which the Remedy e-mail notifications are sent. This must be the same user name you identified when creating the outgoing mailbox in the AR System.
Password	Password associated with the user name. Note: Passing a user name and password in an e-mail message presents a possible security exposure. While the e-mail message resides on the POP3 or IMAP server, someone might be able to access this information.
Time Interval (seconds)	How often (in seconds) the Email Listener port is to check for incoming e-mail messages on the POP3 or IMAP server.
Port	Port to use for the mail server. The default for POP3 is 110; the default for IMAP is 143.
Log out after each mail check	For use with IMAP and multithreading only. If you select Yes , Integration Server logs out a read-only thread to the IMAP mail server after checking for mail on that thread. The main read/write thread to the IMAP server remains intact. If you select No , all the read-only threads remain intact. Select Yes if your IMAP server restricts the number of connections it will allow to remain logged in.

7. Fill in the following fields in the **Security** section to identify the user account to use to execute the service that will handle the e-mail notification.

In this field...	Specify...
Run services as user	If you selected Yes in the Require authorization within message field, the Run services as user field remains blank because the Integration Server expects the user name and password to be in the e-mail message. If you selected No in the Require authorization within message field, you must enter the user under which the service is to run on Integration Server.
Require authorization within message	If you select Yes , Integration Server checks for \$user and \$pass parameters in the Subject line of the e-mail message. The user name is the user under which the service is to run on Integration Server. If

In this field...	Specify...
	you select No , you must specify the user in the Run services as user field above.

8. Fill in the following fields in the **Message Processing** section to identify how the Integration Server is to process the e-mail notification.

In this field...	Specify...
Global Service	Service to be executed on Integration Server. This field overrides a service specified in the Subject line of the e-mail message. Specify the service you created to handle the e-mail notification either in this field or in the Default Service field.
Default Service	Service to be executed if the e-mail message does not provide a valid service in the Subject line and the Global Service field is blank. Specify the service you created to handle the e-mail notification either in this field or in the Global Service field.
Send reply email with service output	Whether you want Integration Server to send output generated by the service to the original sender in an e-mail attachment. If the original e-mail contained multiple attachments, the reply contains an equal number of attachments.
Send reply email on error	Whether you want Integration Server to report errors that occurred during service execution to the original sender in the Body portion of an e-mail message.
Delete valid messages (IMAP only)	Whether you want Integration Server to delete a valid e-mail message from the IMAP server after Integration Server has successfully received the email message. This setting helps prevent e-mail messages from accumulating on the IMAP server, possibly affecting disk space and performance. Integration Server always deletes e-mail messages on a POP3 server.
Delete invalid messages (IMAP only)	Whether you want Integration Server to delete invalid e-mail messages from the IMAP server. Invalid e-mail messages are those that experienced errors during processing. This setting helps prevent invalid e-mail messages from accumulating on the IMAP server, possibly affecting disk space and performance. Integration Server always deletes e-mail messages on a POP3 server.
Multithreaded processing (IMAP only)	Whether you want Integration Server to use multiple threads for this port. This setting allows the port to handle multiple requests at once and avoid a bottleneck.

In this field...	Specify...
Number of threads if multithreading turned on	The number of threads Integration Server is to use for this port. The default is 3.
Invoke service for each part of multipart message	Whether Integration Server should invoke the service for each part of a multipart message or just once for the entire message. You must specify Yes to have Integration Server treat each part of the message individually. That is, Integration Server sends each part to the content handler and then to the specified service.
Include email headers when passing message to content handler	Whether Integration Server includes the e-mail headers when passing an e-mail message to the content handler. Because you select Yes for Invoke service for each part of multipart message , you should select No for this option because each section has its own headers that the content handler and/or the service already knows how to process.
Email body contains URL encoded input parameters	How Integration Server treats input parameters it finds in e-mail messages. When this field is set to Yes , Integration Server considers a string such as ?one=1+two=2 to be a URL-encoded input parameter. It then decodes this string into an IData object, puts it into the pipeline, and passes it to the service. When this field is set to No , Integration Server treats the string as plain text and passes it to the appropriate content handler.

Setting Up Web Services

You can set up the AR System to send notification by invoking a Web service to which it passes an XML document that contains the notification information. When the filter you set up on the AR System detects its condition, it reads the WSDL file you identify in the filter. The WSDL file contains the URL of the Web service. The AR System sends the XML document to the URL that is contained in the WSDL file. The XML document contains the values of the fields you selected when you created the filter.

You create a Web service on Integration Server to process the incoming XML document. After creating the service, you can generate the WSDL that you will need to specify in the filter. You can generate the WSDL from Designer or Digital Event Services. The WSDL you generate must use the SOAP-RPC or SOAP-MSG protocol.

Note:

For more information about using the Web service features in Designer, see the *webMethods Service Development Help* for your release. For information about using Digital Event Services to create Web services, see the *Web Services Developer's Guide* version 8.0. Note that some Web services features available in Designer are not available in Digital Event Services.

For an overview of how the AR System notifies Integration Server via a Web service, see [“Handling Notifications via a Web Service” on page 14](#).

The following table lists the actions you must take in Integration Server to set up Web service notifications and handling them. For information about the actions you must take in the Remedy AR System, see the Remedy documentation.

Step	Tool	Action
1	Designer or Digital Event Services	Create a service that performs the appropriate actions to handle a notification.
2	Designer or Digital Event Services	Generate a WSDL for the service that handles the notification.

Creating a Web Service to Handle the Notification

You create a Web service, which the AR System invokes to notify Integration Server of a change in the AR System.

When AR System invokes the Web service, it passes an XML document that contains information about the change in the AR System. Your Web service should inspect the XML document and take appropriate actions. For example, you might need to update another system in response to the change in the AR System.

Remedy Adapter comes with a sample Web service that you can use to model your own Web service. The sample Web service is `wm.adapter.wmremedy.outbound.webService.receiveWSDL_SOAP_RPC`.

The sample `receiveWSDL_SOAP_RPC` service receives an XML document and performs simple processing. The input signature of the `receiveWSDL_SOAP_RPC` service identifies the set of fields that should be present in the XML document. This means that when the filter for the Web service is configured in the AR System, these required input fields must be mapped in the input mapping of the filter. Additionally, the output signature of the `receiveWSDL_SOAP_RPC` service contains the set of fields that need to be mapped in the output mapping of the filter. The `receiveWSDL_SOAP_RPC` service extracts a few fields from the XML document and changes their values. Then the XML document with the updated field values is sent back to the AR System.

Generating a WSDL for the Service that Handles the Notification

After you create the service and it resides in the Integration Server namespace, you can generate the corresponding WSDL file from Designer or Digital Event Services.

Note:

For more information about using the Web service features in Designer, see the *webMethods Service Development Help* for your release. For information about using Digital Event Services to create Web services, see the *Web Services Developer's Guide* version 8.0. Note that some Web services features available in Designer are not available in Digital Event Services.

7 Predefined Health Indicator

■	Predefined Health Indicator	82
---	-----------------------------------	----

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*.

8 Administrator APIs

■ Administrator APIs	84
----------------------------	----

Administrator APIs

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

9 Configuration Variables Templates for Adapter Assets in Microservices Runtime

■ Configuration Variables Templates for Adapter Assets in Microservices Runtime	86
---	----

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*.

10 Adapter Logging and Exception Handling

■ Overview	88
■ Adapter Logging Levels	88
■ Configuring Adapter Logging Levels	88
■ Adapter Message Logging	89
■ Adapter Exception Handling	90
■ Adapter Error Codes	92

Overview

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

Adapter Logging Levels

Remedy Adapter uses webMethods Integration Server logging mechanism to log messages. You can configure and view Integration Server logs to monitor and troubleshoot Remedy Adapter. 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 Audit Logging Guide* for your release.

Configuring Adapter Logging Levels

Beginning with Integration Server 7.1.1, you can configure different logging levels for Remedy Adapter. For complete information about specifying the amount and type of information to include in the log, see the *webMethods Audit Logging Guide* for your release.

Accessing the Adapter's Logging Information

> 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 Settings for the Adapter

> To change logging settings for the adapter

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

Configuring Adapter Logging Levels

Beginning with Integration Server 7.1.1, you can configure different logging levels for Remedy Adapter. For complete information about specifying the amount and type of information to include in the log, see the *webMethods Audit Logging Guide* for your release.

Accessing the Adapter's Logging Information

> 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 Settings for the Adapter

> To change logging settings for the adapter

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

Adapter Message Logging

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

The following table lists the logging levels when you are running Remedy Adapter on Integration Server 7.1.1 or higher:

Integration Server	Log	Description
Integration Server 7.1.1 or 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 Remedy Adapter service template on the Audit tab.
	Error Log	Remedy Adapter automatically posts fatal-level and error-level log messages to the error log. These log messages appear as adapter run-time messages.

Integration Server	Log	Description
	Server Log	Remedy Adapter 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 Remedy Adapter log messages.

The Remedy Adapter's log messages appear in the format `ADA.0760.nnnnc`, where:

- `ADA` is the facility code that indicates the message is from an adapter.
- `0760` (or `760`) is the Remedy Adapter major error code, which indicates that the message is generated by Remedy Adapter.
- `nnnn` represents the error's minor code. For detailed descriptions of the Remedy Adapter's minor codes, see [“Adapter Error Codes” on page 92](#).
- `c` represents the critical level of the error. For Remedy Adapter this will be either `V2-Verbose2` or `V4-Verbose4`. Beginning with Integration Server 7.1.1, all verbose logging levels appear as Trace logging level.

Important:Remedy Adapter logs all messages at Trace logging level.

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

- 0113 Adapter Runtime (Managed Object)
- 0114 Adapter Runtime
- 0117 Adapter Runtime (Adapter Service)
- 0118 Adapter Runtime (Connection)
- 0121 Adapter Runtime (SCC Transaction Manager)
- 0126 Adapter Runtime (SCC Connection Manager)

Adapter Exception Handling

Remedy Adapter throws the following exception classes that you should be aware of as you build integrations using the adapter:

- `AdapterException` is thrown when an `ARException` is thrown in the AR System. For more information, see [“AdapterException” on page 91](#).
- `AdapterConnectionException` is thrown when a fatal error occurs and the AR System becomes unavailable. For more information, see [“AdapterConnectionException” on page 91](#).

In all cases, the adapter passes the underlying exception to the adapter runtime, which wraps it in a container exception that it then passes to Integration Server. Integration Server then serializes

the exception and returns it to the client service. Typically, that client (for example, a flow or Java service that calls an adapter service) will include logic that traps these exceptions and branches accordingly. For information about how to trap the exception in a flow, see the *webMethods Service Development Help* for your release.

AdapterException

When an `ARException` occurs in the AR System, the exception is wrapped in an `AdapterException`. An `AdapterException` is for a non-fatal errors where the AR System is still available. For example, an `AdapterException` would be thrown when an error occurs while trying to create an entry in the AR System. The AR System's error message is logged in the `server.log` file. The logged error message includes the Remedy message and the reference to the field ID and the field name.

When an `AdapterException` is thrown, the connection is not dropped. Remedy Adapter only drops connections for fatal errors that result in an `AdapterConnectionException`. If you determine that there are AR System API call errors for which you want the connection dropped, you can configure Remedy Adapter to identify additional error codes that should be considered fatal. For more information about `AdapterConnectionExceptions`, see [“AdapterConnectionException” on page 91](#) below. For more information about configuring error codes that should be considered fatal, see [“Configuring Additional Fatal Error Codes” on page 91](#).

AdapterConnectionException

An `AdapterConnectionException` is thrown when a fatal error occurs and the AR System becomes unavailable, for example due to a network problem or when the AR System Server is shut down.

Remedy Adapter recognizes a fatal error based on the AR System API call error code. An example of an error that Remedy Adapter considers fatal is error code 90, which indicates that the AR System catalog could not be opened.

When an `AdapterConnectionException` is thrown, WmART drops the connection from the connection pool and tries to create a new connection. WmART then wraps the exception in `com.wm.pkg.art.error.DetailedSystemException` and throws it to the Integration Server.

In addition to the API call error codes that Remedy Adapter considers fatal, you can configure additional error codes that you want Remedy Adapter to treat as fatal; that is, additional error codes for which you want Remedy Adapter to drop the connection and throw an `AdapterConnectionException`. For more information, see [“Configuring Additional Fatal Error Codes” on page 91](#) below.

Configuring Additional Fatal Error Codes

Remedy Adapter recognizes a set of AR System API call error codes as fatal. These error codes indicate that the AR System has become unavailable. In response to a fatal error code, Remedy Adapter throws an `AdapterConnectionException`, drops the connection from the connection pool, and tries to create a new connection.

You can configure Remedy Adapter to add to the list of error codes that it considers fatal. As a result, when Remedy Adapter receives one of the fatal error codes you configure, it behaves in

the same manner; that is, throws an `AdapterConnectionException`, drops the connection from the connection pool, and tries to create a new connection.

To identify additional fatal error codes, use the Integration Server Administrator to configure the `watt.adapter.Remedy.fatalErrors` server configuration parameter in the Extended Settings. You use this configuration parameter to list the AR System API call error codes that should be considered fatal.

When an AR System API call results in an error and an `ARException` is thrown, Remedy Adapter catches the exception. Remedy Adapter examines the error code and if it matches a code listed on the `watt.adapter.Remedy.fatalErrors` configuration parameter, an `AdapterConnectionException` is thrown rather than a `AdapterException`.

➤ To configure additional fatal error codes

1. Start your Integration Server and the Integration Server Administrator if they are not already running.
2. From the Integration Server Administrator, from the **Settings** menu in the navigation panel, select **Extended**.
3. Click **Edit Extended Settings**.
4. In the Extended Settings box, add a line for the `watt.adapter.Remedy.fatalErrors` server configuration parameter.

For example, if you want Remedy Adapter to consider error codes 91, 95, and 99, as fatal, you would use the following:

```
watt.adapter.Remedy.fatalErrors=+91, 95, 99
```

Note:

By default error code number 90 is treated as a fatal error. You can append a new error code to the existing list.

For more information about setting server configuration parameters, see the *webMethods Audit Logging Guide* for your release.

5. Click **Save Changes**.

Adapter Error Codes

This section lists the Remedy Adapter's minor codes and provides information on the message, reason, and possible action for each error.

Remedy Adapter categorizes its minor code numbers as follows:

Error Number Range	Descriptions
0001-0099	Adapter-specific informational logging messages.
0100-0999	Adapter-specific warning messages. For more information, see “Warning Messages” on page 93 .
1000-1999	Adapter-specific error messages. For more information, see “Error Messages” on page 94 .

Important:

The Remedy Adapter logs all messages at the Trace logging level.

Warning Messages

Warning Code	Description
0100	Login Status for userName - message.
	Explanation: The AR System user configured in the Remedy Adapter connection does not exist on the AR System Server. However, the logon succeeded and the connection was enabled with a guest user on the AR System. Note that this happens only when the AR System is configured to allow guest users logon.
	Action: Check the user account you have defined in the adapter connection. If you do not want to allow guest user logon, configure the AR System to deny logon to guest users.
0200	Remedy Adapter cannot retrieve the list of available form names. Check the log for details.
	Explanation: When creating an adapter service, Designer displays appropriate adapter service settings in the Flow Service Editor. Designer accesses logic in Remedy Adapter to obtain the appropriate settings for the adapter service. Remedy Adapter was unable to obtain all the form names from the AR System. As a result, the Form Name field in the adapter settings will not list all available forms.
	Reason: Remedy Adapter is unable to obtain all form names might be that the user does not have the privilege to access one or forms, or there might be a problem with one or more forms.
	Action: If the Form Name field contains the form you want to use, proceed. Otherwise, use have an AR System administrator check the form and its accessibility, and make corrections as necessary.

Warning Code	Description
0300	Remedy field type [Field Type Identifier] could not be mapped to a data type supported by the Remedy Adapter; data type java.lang.Object will be used.
	Explanation: Remedy Adapter is retrieving a value from an AR System field. However, the data type of the AR System field is not supported by Remedy Adapter. Remedy Adapter will use the data type java.lang.Object for the field. For more information about data type mappings, see “AR System Data Type to Adapter Data Type Mapping” on page 100 .
	Action: To avoid the error, you can update the Input Fields tab of the adapter service to unselect the Use Field check box.

Error Messages

Error Code	Description
1001	Problem connecting to AR system - user: userName.
	Explanation: Remedy Adapter failed to establish a connection to an AR System Server. Remedy Adapter attempted to establish the connection using the specified user. Previous messages display information about the AR System Server to which Remedy Adapter is attempting to make a connection. An AdapterConnectionException is thrown.
	Action: Ensure that the AR System Server is running. If the AR System Server is running, ensure that you have your environment set up correctly. For information about the required AR System jar and dll files, see “Installing Remedy Adapter 7.1” on page 28 .
1002	Problem clearing AR system's userInfo object.
	Explanation: AR System uses a userInfo object to maintain the session state of each of its clients. After Remedy Adapter destroys a connection, it attempts to clear the userInfo object using the userInfo.clear() API. If the userInfo.clear() API results in an ARException being thrown, the exception is wrapped into an AdapterException, and Remedy Adapter issues this message.
	Action: From the Integration Server Administrator, reload the WmRemedyAdapter package. Then from the Remedy Adapter administrative screens, re-enable the adapter connection. For instructions, see “Enabling Adapter Connections” on page 49 .
1003	Invalid value specified for input parameter parameterName. The input cannot be empty or null.
	Explanation: When running a Remedy Adapter service, one of the input values specified for <i>parameterName</i> was invalid.

Error Code	Description
	Action: Check the service <i>parameterName</i> and provide a valid value. Rerun the service.
1005	Empty or null values specified for input.
	Explanation: The values provided for one or more of the parameters of a Remedy Adapter service were invalid.
	Action: Check the service signature and provide valid values. Rerun the service.
1010	Adapter Service Operation Name service execution failed on Form: <i>formName</i> . Error: Message.
	Explanation: Internal error occurred while executing the specified operation/adapter service.
	Action: Check the error message for a possible correction. If there is none, contact Software AG Global Support.
1018	ResourceDomainLookup failed.
	Explanation: Remedy Adapter encountered a problem while attempting to generate adapter service templates.
	Action: From the Integration Server Administrator, reload the WmRemedyAdapter package. Then in Designer, refresh the Integration Server host. If the problem persists, restart Integration Server.
1021	Missing required input(s) for ARS Server Operation: Operation Name.
	Explanation: There are missing input parameters for the specified <i>Operation Name</i> in the ARS Server Operation adapter service.
	Action: Fill in the required input parameters for the service and then rerun the service.
1022	ARS Server Operation Operation Name execution failed.
	Explanation: An exception occurred when running the specified <i>Operation Name</i> in the ARS Server Operation adapter service.
	Action: Ensure that a connection to the ARS server is available. Ensure that all dependent jars and dll are located in the <i>Integration Server_directory \packages\WmRemedyAdapter\code\jars</i> and the <i>Integration Server_directory \lib</i> directory respectively.
1023	Invalid migration package Package Name.
	Explanation: The Remedy Adapter migrate_60_To_71 service failed, because the specified target package was invalid.

Error Code	Description
	Action: Specify a valid package in the <i>packageName</i> parameter of the <i>migrate_60_To_71</i> service and rerun the service.
1100	ARException occurred while trying to create entry on Form: formName. ARException Message.
	Explanation: A Create adapter service was executed to create an entry in the AR System. However, the Create adapter service failed after the field values for the new entry were set.
	Action: Review the input signature of the Create adapter service and ensure that valid values were supplied as input and that the input variables have the correct data types.
1200	ARException occurred while trying to delete entry <<entryID>> on Form: formName. ARException Message.
	Explanation: A Delete adapter service was executed to delete an entry in the AR System. However, the Delete adapter service failed to delete the entry.
	Action: Check the input and output signatures of the adapter service and ensure valid values were supplied as input and that the input variables have the correct data types.
1300	ARException occurred while trying to get entries from Form: formName. ARException Message.
	Explanation: A Get or Batch Get adapter service was executed to retrieve detail information about one or more entries in the AR System. However, the Get or Batch Get adapter service failed after reading the fields in the entry or entries.
	Action: Check the input and output signatures of the adapter service. Ensure the output fields have the correct data types.
1400	ARException occurred while trying to update the entry [entryID] on Form: formName. ARException Message.
	Explanation: An Update adapter service was executed to update information for an entry in the AR System. However, the Update adapter service failed after setting the fields in the entry.
	Action: Review the input signature of the Update adapter service and ensure valid values were supplied as input and that the input variables have the correct data types.
1500	ARException occurred while trying to find entries from Form: formName, using the query [query]. ARException Message.

Error Code	Description
	Explanation: A Find adapter service was executed to retrieve details from the AR System. However, the adapter service encountered an error while getting the list entries for the specified query.
	Action: Check the query and output signature of the Find adapter service.

A Data Mapping

■ Overview	100
■ AR System Data Type to Adapter Data Type Mapping	100
■ Unsupported Remedy Data Types	101

Overview

The appendix provides information about the supported AR System data types and how they map to Remedy Adapter data types. It also explains how Remedy Adapter handles unsupported AR System data types.

AR System Data Type to Adapter Data Type Mapping

The information provided in this section maps all supported AR System data types to the data types used by Remedy Adapter.

Remedy Data Types	Data Type ID	Adapter Data Types
ATTACHMENT	11	<p>com.wm.data.IData (in Create, Update, Get, and Batch Get services)</p> <p>Document with a name that corresponds to the attachment field name in the Remedy form on the Remedy ARS Server. The document (IData) has two fields:</p> <ul style="list-style-type: none"> ■ <i>bytes</i> Byte Maps to attachment data in byte array format. ■ <i>title</i> String Maps to attachment filename.
BITMASK	8	java.util.Integer
BYTES	9	java.lang.Byte[]
CHAR	4	java.lang.String
CURRENCY	12	java.lang.String
DATE	13	java.util.Date
DATE/TIME	7	java.util.Date
DECIMAL	10	java.math.BigDecimal
DIARY	5	<p>java.lang.String[] (in Get service)</p> <p>java.lang.String (in Create and Update services)</p>
ENUM (Selection)	6	java.lang.String
INTEGER	2	java.lang.Integer
KEYWORD	1	java.lang.String
REAL	3	java.lang.Double
TIMESTAMP	7	java.util.Date

Unsupported Remedy Data Types

Remedy Adapter attempts to map AR System data types that it does not support to `java.lang.String`.

Remedy Adapter detects an unsupported data type when it catches a `java.lang.ClassCastException`. No exception is thrown for unsupported data types. Remedy Adapter attempts to map the unsupported data type to `java.lang.String` and logs information about the unsupported data type.

If an unsupported data type cannot be cast as a `java.lang.String`, Remedy Adapter does not retrieve the field value with the unsupported data type from the AR System. As a result, the value will not be returned to the adapter service requesting it. In this situation, Remedy Adapter logs messages to indicate the field could not be retrieved.

B webMethods Remedy Adapter Services

■ Overview	104
■ WmRemedyAdapter Package	104

Overview

This appendix describes the built-in services and other elements, provided with the WmRemedyAdapter package of webMethods Remedy Adapter.

WmRemedyAdapter Package

The WmRemedyAdapter package contains public services used to implement and support the functionality of Remedy Adapter. The following elements are available in this package:

wm.adapter.wmremedy.admin

Contains services that register and unregister Remedy Adapter with Integration Server.

wm.adapter.wmremedy.outbound

Contains services that you use to process the documents that the AR System sends to Integration Server.

wm.adapter.wmremedy.util

Contains the `wm.adapter.wmremedy.util:migrate_60_To_71` service that you use to migrate services created with version 6.0 of Remedy Adapter to version 7.1.

wm.adapter.wmremedy.util:migrate_60_To_71

Migrates services created with Remedy Adapter 6.0 to Remedy Adapter 7.1.

Input

<i>packageName</i>	String Specifies the name of the package containing the services that need to be migrated. All Remedy Adapter services found in this package will be migrated to the same package by modifying their templates to the Remedy Adapter 7.1 templates.
--------------------	--

Output

<i>status</i>	String Specifies the result of the operation. Valid values: <ul style="list-style-type: none">■ SUCCESS all services in the specified package were migrated successfully.■ FAILURE the operation failed. For more information about the error that occurred, see the Integration Server error log.
---------------	--
