



Mercury IT Governance Center™
**Mercury Service Desk Adapter for
Peregrine Configuration Guide**

Version: 1.0

MERCURY™



This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332; 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

© 1997–2006 Mercury Interactive Corporation. All rights reserved.

If you have any comments or suggestions regarding this document, please send email to documentation@mercury.com.

Table of Contents

List of Tables	v
Chapter 1: Introduction.....	7
About This Document.....	8
Who Should Read This Document	8
Related Documents.....	8
About the Service Desk Adapter	9
Supported Applications	9
Change Request Conversion Flow.....	10
Preparing Service Desk Integration.....	11
Adapter Configuration File	11
Conversion Scripts	11
Deployment	12
Chapter 2: Configuring the Adapter.....	13
Configuring the Adapter	14
Location and Naming Conventions of Adapter Configuration Files	14
Setting Up the Adapter File.....	15
Password Encryption.....	16
Configuring the Common Adapter Attributes.....	16
Configuring the Connector Attributes.....	18
Peregrine Connector Settings.....	18
Configuring the Converter Attributes	20
Conversion Script Functions	21
Conversion Script APIs.....	21
Conversion Script Example.....	23
Configuring the Mercury IT Governance Center Sender Settings.....	25

Chapter 3: Configuration File Examples	27
Adapter Configuration File Examples	28
Peregrine Adapter Configuration File Example.....	28
Peregrine Web Services Adapter Configuration File Example.....	28

List of Tables

Table 1-1	Supported applications	9
Table 2-1	Common Adapter attributes.....	17
Table 2-2	Peregrine connector attributes.....	18
Table 2-3	Peregrine Web Services connector attributes.....	19
Table 2-4	Converter attributes.....	20
Table 2-5	Mercury IT Governance Center sender attributes	25

Chapter 1 Introduction

In This Chapter:

- *About This Document*
 - *Who Should Read This Document*
 - *Related Documents*
 - *About the Service Desk Adapter*
 - *Supported Applications*
 - *Change Request Conversion Flow*
 - *Preparing Service Desk Integration*
 - *Deployment*
-

About This Document

This document:

- Describes the Mercury Service Desk Adapter for Peregrine, which provides integration of Mercury IT Governance Center with Peregrine Service Center.
- Provides instructions for deploying the Mercury Service Desk Adapter to multiple servers, if necessary.
- Discusses configuration of the Mercury Service Desk Adapter for Peregrine.

Who Should Read This Document

This document is intended for Mercury IT Governance Center system administrators who are also familiar with Peregrine Service Center.

Related Documents

Related documents for this document include:

- *Mercury Demand Management User's Guide*
- *Mercury Accelerator for IT Service Management Guide*

About the Service Desk Adapter

Mercury IT Governance Center contains a single repository for application change requests. When a user creates a change request on your local service desk system, the request must be imported into Mercury IT Governance Center for processing.

The Service Desk Adapter provides the ability to import requests from service desk systems, such as Remedy AR System and Peregrine Service Center, into Mercury IT Governance Center.

The requests are imported using an Adapter specific to the service desk system you are using. This process is controlled by a scheduler that can be configured to run at a specified time, for example once a day.

Importing change requests into Mercury IT Governance Center is a one-way process — once the change request has been imported from the service desk system into Mercury IT Governance Center, there is no need for Mercury IT Governance Center to send any modifications back to the service desk system, nor to notify the service desk system on the progress of that change request in Mercury IT Governance Center.

Supported Applications

The Service Desk Adapter supports the following applications described in [Table 1-1](#):

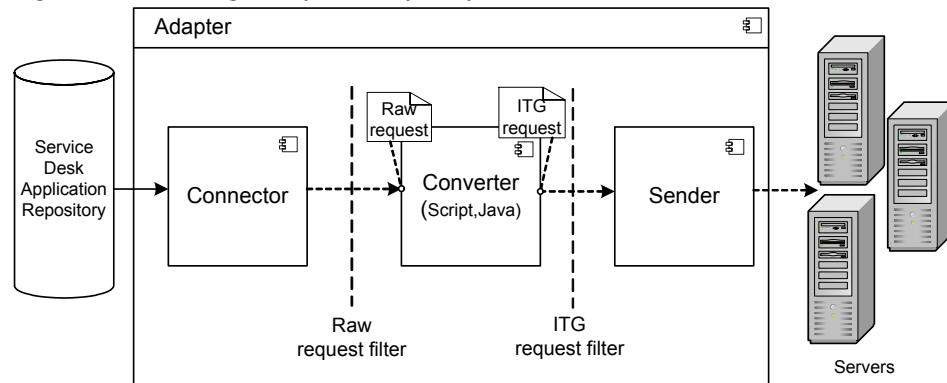
Table 1-1. Supported applications

Application	Version
Remedy Action Request System	5.0
Peregrine Service Center	6.1 This version also supports Peregrine Web Services. Prior versions are supported using the Peregrine Connect-It application.

Change Request Conversion Flow

Figure 1-1 depicts the flow for importing a change request from a service desk system into Mercury IT Governance Center.

Figure 1-1. Change request import process



For each service desk system, there is an Adapter. This Adapter is responsible for the process of importing change requests from the service desk system into Mercury IT Governance Center.

The Adapter is composed of three subcomponents.

- **Connector** — Collects new change requests from the service desk systems.
- **Converter** — Converts the change request from the service desk data model it was created in, into the Mercury IT Governance Center data model.
- **Sender** — Sends the converted change requests to Mercury IT Governance Center.

The Adapter also contains a raw request filter and a Mercury IT Governance Center request filter. Using these filters, you can control which change requests are imported into Mercury IT Governance Center. The raw change request filter filters the information in the service desk data model — that is, before it converts the requests. The Mercury IT Governance Center request filter then filters the information in the Mercury IT Governance Center data model — after conversion, but before it is imported into Mercury IT Governance Center.

Preparing Service Desk Integration

Before you import a change request from your service desk system into Mercury IT Governance Center, you need to configure the Adapter.

To configure the Adapter, you need to:

- Set up the adapter configuration file.
- Write the conversion scripts.

Adapter Configuration File

The adapter configuration file is an XML file that contains the Adapter's general settings, as well as specific attributes that are relevant to the service desk application you are working with. For details about setting up this file, see [Setting Up the Adapter File on page 15](#).

Conversion Scripts

Conversion scripts are called by the adapter configuration file, and are responsible for field mapping during the conversion process.

Each script must contain at least the `Convert`, `preFilter`, and `postFilter` functions, which are described in [Conversion Script Functions on page 21](#). These functions enable you to specify the type of change requests to convert from the service desk data model to the Mercury IT Governance Center data model (for example, requests whose status is NEW) and which requests to import into Mercury IT Governance Center (for example, only requests of a specific type).



Note

You can configure more than one Adapter per service desk system. This enables you to import change requests from several servers of the same service desk system, or import several request types from the service desk system.

Deployment

The Service Desk Adapter is deployed as part of Mercury IT Governance Center version 6.0 Service Pack 9. It can also be deployed on other application servers.

To deploy the Service Desk Adapter from Mercury IT Governance Center on a different application server:

1. Copy the Service Desk Adapter's .war file, `ServiceDesk_Integration.war` (located in `<ITG_Home>/server/<ITG_Instance>/deploy`), to the deploy directory on the new application server.
2. Create the following directories in the home directory (`<Application_Server_Home>`) of the application server:
 - `conf/sdi`
 - `bin/sdi`
 - `security`
3. Copy the content of the Mercury IT Governance Center server's `conf/sdi`, `bin/sdi`, and `security` directories into the respective directories that you just created on the application server.
4. Create a JVM property called `sdi.home`, and set it to point to the new application server's home directory.
5. Restart the application server.

Chapter 2 Configuring the Adapter

In This Chapter:

- *Configuring the Adapter*
 - *Location and Naming Conventions of Adapter Configuration Files*
 - *Setting Up the Adapter File*
 - *Password Encryption*
 - *Configuring the Common Adapter Attributes*
 - *Configuring the Connector Attributes*
 - *Configuring the Converter Attributes*
 - *Configuring the Mercury IT Governance Center Sender Settings*
-

Configuring the Adapter

The Adapter is made up of a connector, a converter, and the Mercury IT Governance Center Sender. The connector connects your service desk system to the Service Desk Adapter. The converter converts the raw change requests from your service desk data model into the Mercury IT Governance Center data model. The Mercury IT Governance Center sender sends — or exports — the converted request to Mercury IT Governance Center.

The Adapter is configured in the adapter configuration file, which is an XML file that contains the Adapter's general settings, the connector's attributes, and the conversion scripts that enable the conversion of a request from the service desk data model into the Mercury IT Governance Center data model.

Location and Naming Conventions of Adapter Configuration Files

The Adapter configuration files are located in the `conf\sdi` folder of the Mercury IT Governance server.

The `conf\sdi` folder contains:

- A configuration file for each Adapter.

Inside the configuration file, you define a name for the Adapter. The name of the configuration file must be identical to the name defined for the Adapter, and must have a `.settings` extension, as follows:

```
<adapter name>.settings
```

For example, if the Adapter name is `peregrine-adapter`, the configuration file name will be `peregrine-adapter.settings`.

- A subfolder for each adapter configuration file. The subfolder holds the conversion scripts responsible for the field mapping and filtering of the requests.

The name of the subfolder must also be identical to the name defined for the Adapter, and must have a `.ext` extension, as follows:

```
<adapter name>.ext
```

Following the example above, the `conf\sdi` folder will contain a subfolder called `peregrine-adapter.ext`, to hold all the script files for the Peregrine Adapter.

Setting Up the Adapter File

The Adapter configuration file is divided into four main sections:

- The Adapter's attributes, such as the Adapter's name, the name of the service desk application in which the change requests were created, and the type of request being converted.

To set the Adapter attributes, see [Configuring the Common Adapter Attributes on page 16](#).

- The connector attributes—each service desk system has its own set of connector attributes that enable the Adapter to connect to the service desk system.

To set the connector attributes, see [Configuring the Connector Attributes on page 18](#).

- The converter attributes, which call the conversion script files where the field mapping and filter functions are defined.

To set the converter attributes, see [Configuring the Converter Attributes on page 20](#).

- The Mercury IT Governance Center sender information—the attributes for sending the converted and filtered change request data to Mercury IT Governance Center.

To set the Mercury IT Governance Center sender attributes, see [Configuring the Mercury IT Governance Center Sender Settings on page 25](#).

Password Encryption

All the passwords in the configuration files should be encrypted using the Mercury IT Governance Center script, `kEncrypt.sh`, which is located in the `bin` directory of the Mercury IT Governance server.



Note

Encrypted passwords, which contain special characters, must be created in a CDATA section in the configuration file. This is because the configuration file is an XML file, and special characters can be added to an XML file in a CDATA section (`<![CDATA[]>`) only.

Configuring the Common Adapter Attributes

The adapter configuration file identifies the Adapter.

```
<adapter adapter-name="<adapter name>">
  <service-desk-application><SD appl></service-desk-
application>
  <request-type><request type></request-type>
  <number-of-tickets><number of tickets></number-of-tickets>
  <polling-schedules><schedule></polling-schedules>
  .
  .
  .
  .
<<adapter adapter-name=
```

[Table 2-1](#) describes the Adapter attributes common to all service desk systems:

Table 2-1. Common Adapter attributes

Property Name	Description	Default Value
adapter-name (mandatory)	A logical name that represents the collector's name on the client machine. For example: peregrine-adapter This name is also used for the adapter configuration (.settings) file, and the scripts (.ext) folder. (See Location and Naming Conventions of Adapter Configuration Files on page 14.)	(none)
service-desk-application (mandatory)	A logical name for the service desk application used. For example: Peregrine Service Center This name is also used for the service desk system name, defined in the configuration of the Mercury IT Governance Center sender (sdSystemFieldName).	(none)
request-type (mandatory)	A logical name of the request type used. For example: Change Request Form	(none)
number-of-tickets	Sets the number of change requests that the collector processes at a time.	50
polling-schedules	A list of cron expressions separated by the newline character. Format: 30 * * * * <new line> 0 * * * *	(none)
polling-frequency	The frequency (in seconds) that the collector polls for change requests.	If polling-schedules and polling-frequency are undefined, then the default is 30 seconds.

Configuring the Connector Attributes

The second section of the adapter configuration file contains the connector attributes. These attributes vary according to the service desk system you are working with.

```

:
:
<connector>
  <connector-type><SD name></connector-type>
  <properties>
    :
    :
    :
    :
  </properties>
</connector>
:
:
:

```

Peregrine Connector Settings

For the Peregrine service desk system, define the connector attributes as defined in [Table 2-2](#):

Table 2-2. Peregrine connector attributes

Property Name	Description	Default Value
connector-type (mandatory)	This must be set to xmlFolderWatcher.	(none)
idPropertyName (mandatory)	The property name of the ID in the XML file.	(none)
lastUpdatePropertyName (mandatory)	The property name of the last updated value in the XML file.	(none)
directoryName (mandatory)	The shared folder directory path.	(none)
pattern	The file name pattern as a regular expression. For more details, see http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html	No pattern; all files will be read.

Peregrine Web Services Connector Settings

For the Peregrine Web Services service desk system, you define the connector attributes as described in [Table 2-3](#):

Table 2-3. Peregrine Web Services connector attributes

Property Name	Description
connector-type (mandatory)	When importing a change from the change management module, this value must be peregrineChange. When importing a task from the change management module, this value must be peregrineTask.
idProperty (mandatory)	The property name of the ID field in the instance returned from the Web service.
lastUpdatedProperty (mandatory)	The property name of the last updated field in the instance returned from the Web service.
timeZone (mandatory)	The Peregrine server time zone. This must have the following format: GMT<+/- X>
wsDateFormatPattern (mandatory)	The date format used in the Web service answer. For available formats see: http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html
queryDateFormatPattern (mandatory)	The date format used for querying the service center system (as used in the UI expert search). For available formats see: http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html
serviceUrl (mandatory)	The Web service URL.
userName (mandatory)	The user name in the service center system.
password (mandatory)	The password in the service center system. The password should be encrypted. See Password Encryption on page 16 .

Generating the Web Services Stub File

In Peregrine Service Center, you can modify the availability of fields through the Web Services. Each time you modify these settings, a new WSDL is created. You need to regenerate the Web Services stub (.jar) file from the new WSDL.

To generate the stub, use the following utility script:

1. Locate the `bin/sdi` directory on the Mercury IT Governance server.
2. Run `kGeneratePeregrineStub.sh <wsdl-uri> <ITG-Server-Name>`.

For example:

```
kGeneratePeregrineStub.sh http://machine:12670/
ChangeManagement?wsdl ITG_Server
```

Configuring the Converter Attributes

The third section of the adapter configuration file contains the converter attributes. The converter maps the fields from the service desk data model to the Mercury IT Governance Center data model, and filters the requests.

```
.
.
<converter>
  <converter-type>bsfConverter</converter-type>
  <properties>
    scripts=.....
  </properties>
</converter>
.
```

The attributes include the type of converter, and the conversion script file names, as described in [Table 2-4](#).

Table 2-4. Converter attributes

Property Name	Description
converter-type (mandatory)	This must be set to bsfConverter.
scripts (mandatory)	A comma-separated list of script file names. These files must reside in the Adapter's extension folder (<code>conf\sdi\ <adapter name>.ext</code>). For an example of a conversion script file, see Conversion Script Example on page 23 .

Note

The current JavaScript engine does not handle long lines. Make sure that the script line does not exceed 256 characters.

Conversion Script Functions

The conversion scripts are responsible for field mapping during the conversion of service desk model requests into Mercury IT Governance Center model requests, and for filtering the requests. Each script must contain at least the `convert`, `preFilter`, and `postFilter` functions.

- **convert** — This function maps the fields of the service desk request to Mercury IT Governance Center fields.

```
convert(rawTicket, ItgRFC)
```

- **preFilter** — This function filters the change requests before they are converted to the Mercury IT Governance Center data model, ensuring that no unnecessary requests are converted.

```
preFilter(rawTicket)
```

- **postFilter** — This function filters the converted change requests, ensuring that only the desired requests will be imported to Mercury IT Governance Center.

```
postFilter(ItgRFC)
```

Note

For details about conversion APIs, see [Conversion Script APIs](#).

Conversion Script APIs

You can use the following syntax for the conversion scripts.

rawTicket Object

The `rawTicket` object represents the service desk model request. Use the following function to retrieve service desk model request data:

```
get(String fieldName);
```

ItgRFC Object

The ItgRFC object represents the Mercury IT Governance Center request. You can modify the Mercury IT Governance Center request data using the API functions as follows:

Reference ID

Use the following function to set the service desk change request ID in the Mercury IT Governance Center request:

```
setRefId(String referenceId);
```

Time Stamp

Use the following function to set the last update time in the Mercury IT Governance Center request:

```
/**
 * Set the time stamp in long format—that is, the number of
 * milliseconds
 * since January 1, 1970, 00:00:00 GMT
 */
setUpdatedTimeStamp(long updatedTimeStamp);
/**
 * Set the time stamp in the simple date format, which is
 * described
 * at the following location:
 * http://java.sun.com/j2se/1.4.2/docs/api/java/text/
 \* SimpleDateFormat.html
 */
setUpdatedTimeStamp(String updatedTimeStamp, String format);
```

Status

Use the following function to change the status of the Mercury IT Governance Center request. This allows the workflow of the request to progress:

```
setStatus(String newStatus)
```

General Field

Use the following function to set a value of a general field in the Mercury IT Governance Center request:

```
set (String fieldName, String value);
```

Date

Use the following function to set a date field value in the Mercury IT Governance Center request:

```
/**
 * Set the date in long format—that is, the number of
 * milliseconds since
 * January 1, 1970, 00:00:00 GMT
 */
setDateValue(String fieldName, long date);
/**
 * Set the date in the simple date format which is described in
 * the following * location:
 * http://java.sun.com/j2se/1.4.2/docs/api/java/text/
 * SimpleDateFormat.html
 */
setDateValue(String fieldName, String date, String format);
```

Conversion Script Example

Below is an example of a converter script that converts a request from Peregrine to a Mercury IT Governance Center data model:

```
// ITG fields
var DESCRIPTION = "REQ.DESCRPTION";
var PRIORITY_NAME = "REQ.PRIORITY_NAME";
var DETAILS = "REQD.REQUIREMENT_DETAILS";

function convert(peregrineRFC, ItgRFC) {
  // Set the ID of the Peregrine ticket. This is a required
  field.
  ItgRFC.setRefId(peregrineRFC.get("header.changeNumber"));

  // Set the updated time stamp of the ticket in peregrine. The
  format of
  // the date in Peregrine is "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'".
  // This is a required field.
  ItgRFC.setUpdatedTimeStamp
    (peregrineRFC.get("sysmodtime"), "yyyy-MM-
  dd'T'HH:mm:ss.SSS'Z'");

  // Set the priority.
  var peregrineRfcPriority =
  peregrineRFC.get("header.priority");
  if (peregrineRfcPriority == 1)
    ItgRFC.set(PRIORITY_NAME, "Critical");
  else if (peregrineRfcPriority == 2)
    ItgRFC.set(PRIORITY_NAME, "High");
  else if (peregrineRfcPriority == 3)
    ItgRFC.set(PRIORITY_NAME, "Normal");
  else if (peregrineRfcPriority == 4)
    ItgRFC.set(PRIORITY_NAME, "Low");
  else
    ItgRFC.set(PRIORITY_NAME, "Normal");

  // Get the description.
```

```
    shortDescription =
    peregrineRFC.get("header.briefDescription");

    // If there is short description, set the description field
    in ITG.
    if (shortDescription != null && shortDescription != "") {
        ItgRFC.set(DESCRIPTION,shortDescription);
    }

    longDescription = null;
    longDescriptionContainer =
    peregrineRFC.get("descriptionStructure.description");
    if (longDescriptionContainer != null){
        longDescription = descriptionArrayToString
        (peregrineRFC.get(
"descriptionStructure.description.description"));
    }

    if (longDescription != null) {
        ItgRFC.set(DETAILES,longDescription);
    }
}

function preFilter(peregrineRFC) {
    var status = peregrineRFC.get("header.status");
    if (status != closed) {
        return true;
    }
    return false;
}

function postFilter(ItgRFC) {
    var priority = ItgRFC.get("PRIORITY_NAME");
    if (priority != "Normal" || priority != "Low") {
        return false;
    }
    return true;
}

// Parses an array of description into string (one level parsing
// only - not
// recursive).
function descriptionArrayToString(objList){
    str= "";
    if (objList!=null){
        for(var i = 0; i < objList.length - 1; i++){
            str += objList[i] + "\n";
        }
        str += objList[i];
    }
    return str;
}
```


Configuring the Mercury IT Governance Center Sender Settings

The fourth section of the adapter configuration file contains the Mercury IT Governance Center sender attributes, whereby the converted change request is exported to Mercury IT Governance Center.

Below is an example of the Mercury IT Governance Center sender settings for any service desk system:

```
<sender>
  <sender-type>ITGSender</sender-type>
  <properties>
    serverUrl=http://machine:8080
    userName=admin
    password=<![CDATA[ Encrypted Password ]]>
    requestType=
      Service Desk Integration Request
    updateRequest=false
    ticketIdFieldName=REQD.SD_TICKET_ID
    sdSystemFieldName=REQD.SD_SYSTEM_NAME
  </properties>
</sender>
</adapter>
</settings>
```

Set the attributes as described in [Table 2-5](#):

Table 2-5. Mercury IT Governance Center sender attributes

Property Name	Description	Default Value
sender-type (mandatory)	This must be set to ITGSender.	(none)
serverUrl (mandatory)	The URL of the Mercury IT Governance server. For example, http://machine:8080.	(none)
userName (mandatory)	The user name of the Mercury IT Governance server.	(none)
password (mandatory)	The password of the Mercury IT Governance server. The password should be encrypted. See Password Encryption on page 16 .	(none)
requestType (mandatory)	The Mercury IT Governance Center request type that should be created for the change requests that are being imported.	(none)

Table 2-5. Mercury IT Governance Center sender attributes

Property Name	Description	Default Value
updateRequest	Ensures that modifications made to service desk change request are updated in the Mercury IT Governance Center request. The default value is <code>false</code> . Change requests should be modified in Mercury IT Governance Center only.	false
ticketIdFieldName (mandatory)	The field, in Mercury IT Governance Center, containing the service desk change request ID.	(none)
sdSystemFieldName (mandatory)	The field, in Mercury IT Governance Center, containing the service desk system name.	(none)

Chapter

3

Configuration File Examples

In This Chapter:

- *Adapter Configuration File Examples*
 - *Peregrine Adapter Configuration File Example*
-

Adapter Configuration File Examples

The following adapter configuration file examples for Peregrine and Peregrine Web Services systems can be used as references, or copied and modified to fit your organization's needs.

Peregrine Adapter Configuration File Example

The following code is an example of the adapter configuration file for the Peregrine service desk system:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <adapter adapter-name="peregrine-folder-adapter" >
    <service-desk-application>Peregrine</service-desk-
application>
    <request-type>Change Request Form</request-type>
    <number-of-tickets>50</number-of-tickets>
    <polling-schedules>0/30 * * * * ?</polling-schedules>
    <connector>
      <connector-type>xmlFolderWatcher</connector-type>
      <properties>
        idPropertyName=@Number
        lastUpdatePropertyName=LastUpdatedOn
        directoryName=z:\
        pattern=RFCsDst_.*\*.xml
      </properties>
    </connector>
    <converter>
      <converter-type>bsfConverter</converter-type>
      <properties>
        scripts=convert.js
      </properties>
    </converter>
    <sender>
      <sender-type>ITGSender</sender-type>
      <properties>
        serverUrl=http://wish:9090
        userName=admin
        password= Encrypted Password 
        requestType=
          Service Desk Integration Request
        updateRequest=false
        ticketIdFieldName=REQD.SD_TICKET_ID
        sdSystemFieldName=REQD.SD_SYSTEM_NAME
      </properties>
    </sender>
  </adapter>
</settings>
```

Peregrine Web Services Adapter Configuration File Example

The following code is an example of the adapter configuration file for the Peregrine Web Services service desk system:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <adapter adapter-name="peregrine-adapter" version="2.0">
    <service-desk-application>Peregrine Service Center
    </service-desk-application>
    <request-type>Change Request Form</request-type>
    <number-of-tickets>50</number-of-tickets>
    <polling-schedules>0/30 * * * * ?</polling-schedules>
    <connector>
      <connector-type>peregrineChange</connector-type>
      <properties>
        timeZone=GMT+4
        wsDateFormatPattern=yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
        queryDateFormatPattern=MM/dd/yy HH:mm:ss
        idProperty=header.changeNumber
        lastUpdatedProperty=sysmodtime
        serviceUrl=http://labmlbto01:12670/scsserver61/ws
        userName=falcon
        password=<![CDATA[ Encrypted Password ]]>
      </properties>
    </connector>
    <converter>
      <converter-type>bsfConverter</converter-type>
      <properties>
        scripts=convertChange.js
      </properties>
    </converter>

    <sender>
      <sender-type>ITGSEnder</sender-type>
      <properties>
        serverUrl=http://wish:9090
        userName=admin
        password=<![CDATA[ Encrypted Password ]]>
        requestType=
          Service Desk Integration Request
        updateRequest=false
        ticketIdFieldName=REQD.SD_TICKET_ID
        sdSystemFieldName=REQD.SD_SYSTEM_NAME
      </properties>
    </sender>
  </adapter>
</settings>
```

