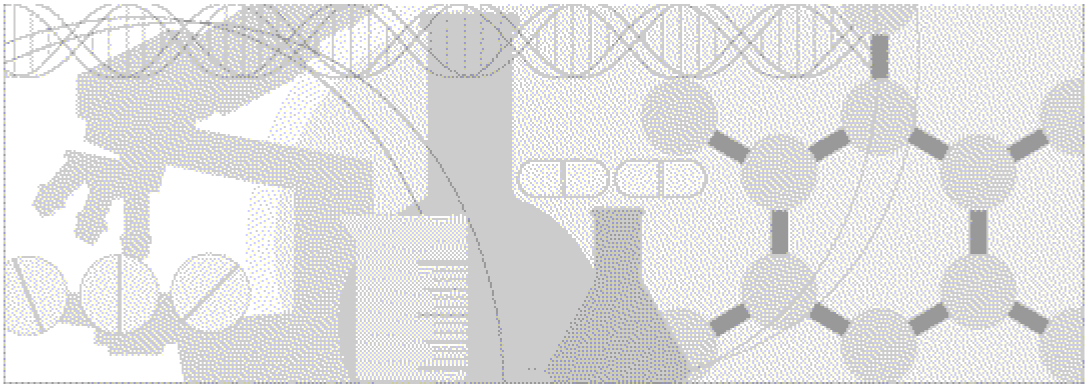
User Guide

Oracle[®] Health Sciences InForm User Management Interface
Release 1.0.4



Part number: E53490-01

Contents

About this guide vii

Chapter 1 User Management Interface overview 1

Chapter 2 Installation 7

Chapter 3 Deployment 17

Chapter 4 Programming model and conventions 23

Chapter 5 Operations 33

Chapter 6 Use Cases 59

About this guide

In this preface

Overview of this guide

The *User Guide* provides an overview of the InForm User Management Interface software including an overview of the software features; details on the system requirements, installation, and deployment; and an API reference that describes the information needed to define InForm study users, sites, authentication, and access privileges programmatically without requiring you to take an InForm study offline.

Audience

This guide is for system and database administrators who are responsible for installing and configuring the User Management Interface software, as well as IT developers responsible for writing the software code to be used by a web service toolkit to communicate with the InForm application.

Documentation

The product documentation is available from the following locations:

- **Oracle Software Delivery Cloud** (https://edelivery.oracle.com)—The complete documentation set.
- **My Oracle Support** (https://support.oracle.com)—*Release Notes* and *Known Issues*.
- **Oracle Technology Network** (http://www.oracle.com/technetwork/documentation)—The most current documentation set, excluding the *Release Notes* and *Known Issues*.

All documents may not be updated for every User Management Interface release. Therefore, the version numbers for the documents in a release may differ.

Title	Description
Release Notes	The *Release Notes* document lists system requirements, describes enhancements introduced and problems fixed in the current release, upgrade considerations, release history, and other late-breaking information.
Known Issues	The *Known Issues* document provides detailed information about the known issues in this release, along with workarounds, if available.
User Guide	The *User Guide* provides an overview of the InForm User Management Interface software including an overview of the software features; details on installation and deployment; and an API reference that describes the information needed to define InForm trial users, sites, authentication, and access privileges programmatically without requiring you to take an InForm trial offline.
Secure Configuration Guide	The *Secure Configuration Guide* provides an overview of the security features provided with the InForm User Management Interface application including details about the general principles of application security and how to install, configure, and use the application securely.
Third Party Licenses and Notices	The *Third Party Licenses and Notices* document includes licenses and notices for third party technology that may be included with the User Management Interface software.

Documentation accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

If you need assistance

Oracle customers have access to support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info, or if you are hearing impaired, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs.

Finding User Management Interface information and patches on My Oracle Support

The latest information about the User Management Interface application is on the Oracle Support self-service website, My Oracle Support. Before you install and use the User Management Interface application, check My Oracle Support for the latest information, including *Release Notes* and *Known Issues*, alerts, white papers, bulletins, and patches.

Creating a My Oracle Support account

You must register at My Oracle Support to obtain a user name and password before you can enter the site.

1. Open a browser to https://support.oracle.com.
2. Click the **Register** link.
3. Follow the instructions on the registration page.

Finding information and articles

1. Sign in to My Oracle Support at https://support.oracle.com.
2. If you know the ID number of the article you need, enter the number in the text box at the top right of any page, and then click the magnifying glass icon or press **Enter**.
3. To search the knowledge base, click the **Knowledge** tab, and then use the options on the page to search by:
   - Product name or family.
   - Keywords or exact terms.

Finding patches

You can search for patches by patch ID or number, product, or family.

1. Sign in to My Oracle Support at https://support.oracle.com.
2. Click the **Patches & Updates** tab.
3. Enter your search criteria and click **Search**.
4. Click the patch ID number.

   The system displays details about the patch. You can view the Read Me file before downloading the patch.
5. Click **Download**, and then follow the instructions on the screen to download, save, and install the patch files.

Finding Oracle documentation

The Oracle website contains links to Oracle user and reference documentation. You can view or download a single document or an entire product library.

Finding Oracle Health Sciences documentation

For Oracle Health Sciences applications, go to the Oracle Health Sciences Documentation page at http://www.oracle.com/technetwork/documentation/hsgbu-clinical-407519.html.

> **Note: Always check the Oracle Health Sciences Documentation page to ensure you have the most up-to-date documentation.**

Finding other Oracle documentation

1 Do one of the following:

- Go to http://www.oracle.com/technology/documentation/index.html.
- Go to http://www.oracle.com, point to the **Support** tab, and then click **Product Documentation**.

2 Scroll to the product you need, and click the link.

Finding prerequisite software for Oracle Health Sciences applications

Prerequisite software for Oracle Health Sciences applications is available from the following locations:

- Download the latest major or minor release from the Oracle Software Delivery Cloud (https://edelivery.oracle.com/).

  For information on the credentials that are required for authorized downloads, click **FAQs** on the main page of the Oracle Software Delivery Cloud portal.

- Download subsequent patch sets and patches from My Oracle Support (https://support.oracle.com).

  To find patch sets or patches, select the **Patches & Updates** tab.

If a previous version of prerequisite software is no longer available on the Oracle Software Delivery Cloud, log a software media request Service Request (SR). Previous versions of prerequisite software are archived and can usually be downloaded. After you open an SR, you can check its status:

- US customers: Call 1-800-223-1711.
- Outside the US: Check www.oracle.com/us/support/contact/index.html for your local Oracle Support phone number.

For more information on logging a media request SR, go to My Oracle Support for Document 1071023.1: Requesting Physical Shipment or Download URL for Software Media (https://support.oracle.com/epmos/faces/DocumentDisplay?id=1071023.1).

CHAPTER 1

User Management Interface overview

In this chapter

About the User Management Interface software

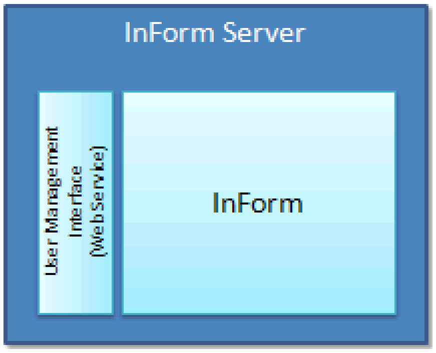
The InForm User Management Interface software is a developer tool kit that provides a web service application programming interface (API) used to perform InForm user and site management tasks without requiring you to take an InForm study offline.

You can use the User Management Interface software to define InForm study users, sites, authentication, and access privileges. The software supports the following functionality in the InForm application:

- Creation and modification of user details, passwords, and activation states.
- Site creation, modification, and user association.
- Group creation, modification, and user association.
- Retrieval of key user data.
- Secure deployment across the Internet.

Architecture

The User Management Interface software is a SOAP-based web service API installed on the InForm application server that provides services for the trials on that server. The API leverages InForm business logic and database tables to perform all operations.



The User Management Interface software uses MedML as the XML schema for representing the provisioning data; that is; the user, site, group, role, and rights data that the software manages. For more information about the MedML format, see the InForm *Utilities Guide*.

Deployment

The User Management Interface software supports network access directly to the server, network access through a load balancer, and local access. After you run the installation, you run a script at the command line, in which you state the type of deployment you want to use. For more information, see ***Deployment types*** (on page 18).

By default, the User Management Interface software is deployed beneath a top-level virtual directory on the web server. For more information, see ***Default deployment*** (on page 18).

For example:

```
https://<myserver>/sdk/provisioning/UserProvisioningService.svc
```

To provide a single point of study management, the User Management Interface software can be deployed directly beneath a study URL using the suffix **/sdk/provisioning** through the use of a network load balancer. For more information, see ***Load balanced deployment*** (on page 19).

For example:

```
https://<myserver>/trialX/sdk/provisioning/UserProvisioningService.svc
```

Scalability

The User Management Interface software is optimized for processing small volumes of data at low rates (bursts of tens of thousands of objects per InForm server per day). Because the software resides on the InForm server, response times will vary based on server load. Service requests will be limited in total processing time as well as request size. For more information, see ***How requests are processed*** (on page 24).

Security

The User Management Interface software uses Secure Sockets Layer (SSL) to provide message encryption and tamper protection of web service calls. Authentication is performed by including a user name and password in the SOAP header. For more information, see ***Writing requests*** (on page 25). The specified credentials must match a pre-established InForm *integration user* that must be pre-defined in each deployed study.

The integration user:

- Is an InForm user whose credentials are included in each request.
- Is created in the InForm application and assigned a password.
- Should not be assigned to any sites or groups.

The credentials for the integration user:

- Are validated against the InForm database and the request is rejected when invalid credentials are provided.
- Are subject to the same password policies (for example, length, account disable, and so on) as InForm end-user accounts, with the exception of password expiration.
- Can be renewed programmatically through the User Management Interface software while still valid.

The User Management Interface software ignores expired account status for valid passwords on active accounts. This behavior minimizes operational overhead that is associated with maintaining individual password expiration windows for the integration user account in each study. As a result, password expiration policy must be implemented through a business operating procedure.

If an invalid password for the integration account is repeatedly entered, the account is deactivated and subsequent calls to the API are rejected. To reactivate the account, the administrator must use the InForm user interface.

CHAPTER 2
Installation

In this chapter

Prerequisite checks

Before you start the installation, make sure that all prerequisites have been met.

For more information, see the system requirements in the *Release Notes*.

The InForm integration user

Before you can use the User Management Interface, you must pre-define an InForm integration user in each deployed trial.

The name and password of the InForm integration user must be included in the SOAP header portion of requests to User Management Interface operations.

The integration user:

- Is an InForm user whose credentials are included in each request.
- Is created in the InForm application and assigned a password.
- Should not be assigned to any sites or groups.

For information about how to create the integration user, see the InForm documentation.

For information about the SOAP header, see ***Writing Requests*** (on page 25).

Installing the User Management Interface software

A silent installation is also supported. For more information, see ***Silent mode*** (on page 13).

1 Download the InForm User Management Interface software from the Oracle Download Center, and extract the ISO.

2 Navigate to the location of the installation files on the downloaded ISO image, and double-click **setup.exe**.

The Preparing Setup progress page appears. When setup is complete, the Welcome page appears.

3 Click **Next**.

The Required System Components page appears.

The installation wizard verifies that the server includes all the necessary software components. Check the listing on this page to determine if any components are missing. Correct any discrepancies before you continue the installation. All required system components must be installed to continue the installation.

4 Click **Next**.

The License Agreement page appears. Review the license agreement carefully. You must accept the terms of the license agreement to continue the installation. Click **Print** to print a copy of the license agreement for your records.

5 Click **Next**.

The Choose Destination Location page appears.

The installer prompts for the installation folder, defaulting to a subdirectory beneath the InForm bin directory.

- Accept the default location (C:\Program Files\Oracle\InForm SDK\Provisioning),

  or

- Click **Change** and browse to the desired location.

The installer automatically adds \Provisioning to the end of whatever directory you select.

6 Click **Next**.

The Ready to Install the Program page appears.

7 Click **Next**.

The files are copied to the server. In addition to copying files to the installation folder, the installer performs the following additional actions:

- Sets the RPC Service and RPC Locator Service to the automatic startup setting, and issues a command to start them on the system in case they were not already started.
- Creates virtual directories and SDK and SDK\Provisioning directories.

If the installer detects that this is an InForm 6.0 or later installation, it prompts for the InForm machine account password. You must enter the password to complete the installation.

When the installation completes, the InstallShield Wizard Complete page appears.

8 Click **Finish**.

Verifying the installation

> **Note:** You must choose the deployment type before verifying the installation. For more information, see ***Deployment types*** (on page 18).

To verify the installation:

- Check IIS to make sure that the sdk and sdk\provisioning virtual directories have been successfully created.
- If you are using the Default deployment, ensure that you have configured the SSL certificate for the sdk\provisioning virtual directory. For more information, see ***Default deployment*** (on page 18).
- Retrieve the web service help page for the User Management Interface. This procedure can be used to determine connectivity, especially when the provisioning service is hosted behind an F5 load balancer.

  For example, to retrieve the help page for trial **rdinformprov004**, browse to

  ```
  https://webexpress.phaseforward.com/rdinformprov004/sdk/provisioning/UserProv
  isioningService.svc?
  ```

Provisioning service that is hosted behind an F5 load balancer

If the provisioning service is hosted behind a load balancer, the URLs sent by the client will include the following:

- The external host name instead of the internal host name.
- The study name that was sent in the original request.

Uninstalling the User Management Interface software

1 Select **Start > Control Panel > Add or Remove Programs**.

2 Select **InForm Provisioning SDK**, and click **Remove**.

A confirmation dialog box appears.

3 Click **Yes**.

All components are removed. Changes to services and permissions do not revert to original settings.

Silent mode

A silent installation and uninstallation are supported. The product ISO contains InstallShield silent response files for installing and uninstalling the InForm User Management Interface, and scripts for running the files.

Silent installation

The file setup.iss contains the pathname of the installation location.

1. Download the InForm User Management Interface software from the Oracle Download Center, and extract the ISO.
2. Navigate to the location of the installation files on the downloaded ISO image.
3. Locate the file setup.iss. If necessary for your configuration, use a text editor to change the values of the following variables in the setup.iss file.
4. Save and close the file.

Variable	Description
SilentInstallLocation	Specifies the location where the InForm User Management Interface will be installed. Default: `C:\Program Files\Oracle\InForm SDK\Provisioning`
SilentInstallLogOn	Specifies whether a log is written for the silent installation. Values: TRUE, FALSE. Default: TRUE.
SilentInstallLogPath	Specifies the complete path to the location of the silent installation log. If the directory does not exist, the silent installation program creates it. Default: `C:\temp\ProvisioningSilentInstallLogs`
InFormAcctPwd	Password for the InForm machine account. There is no default. You must set this value.

5. To begin the silent installation, double-click the file **silent_install.vbs**.

   The InForm User Management Interface is installed.

Default setup.iss file

```
[InstallShield Silent]
Version=v7.00
File=Response File
[File Transfer]
OverwrittenReadOnly=NoToAll
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-DlgOrder]
Dlg0={757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdWelcome-0
Count=7
Dlg1={757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdShowInfoList-0
Dlg2={757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdLicense2Rtf-0
Dlg3={757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdAskDestPath2-0
Dlg4={757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdStartCopy2-0
Dlg5={757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-EnterPassword-0
Dlg6={757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdFinish-0
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdWelcome-0]
Result=1
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdShowInfoList-0]
Result=1
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdLicense2Rtf-0]
Result=1
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdAskDestPath2-0]
szDir=C:\Program Files\Oracle\InForm SDK
Result=1
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdStartCopy2-0]
Result=1
[Application]
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SetupType2-0]
SilentInstallLocation=C:\Program Files\Oracle\InForm SDK\Provisioning
SilentInstallLogOn=TRUE
SilentInstallLogPath=C:\temp\ProvisioningSilentInstallLogs
InFormAcctPwd={CHANGEME}
Result=304
Name=InForm User Management Interface
Version=1.00.0000
Company=Oracle
Lang=0009
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-EnterPassword-0]
szMsg=
Result=1
[{757CC041-B7B1-4BBA-8368-E519A1D3C6B2}-SdFinish-0]
Result=1
bOpt1=0
bOpt2=0
```

Silent uninstallation

1 Navigate to the location of the installation files on the downloaded ISO image.

2 Double-click the file **silent_uninstall.vbs**.

The uninstall.iss file runs, uninstalling the InForm User Management Interface.

If the InForm software is reinstalled on a system with the User Management Interface

If you reinstall the InForm software on a system on which the User Management Interface software is installed, you must update the following registry key after the InForm software installation has been completed.

Registry key for InForm versions 4.6, 5.0, and 5.5:

HKEYHKEY_LOCAL_MACHINE\SOFTWARE\Phase Forward\AuthenticationFilter\ByPassKeyPhrase

Value: sdk/provisioning

Registry key for InForm version 6.0 and above:

HKEYHKEY_LOCAL_MACHINE\SOFTWARE\OracleHS\AuthenticationFilter\ByPassKeyPhrase

Value: sdk/provisioning

CHAPTER 3

Deployment

In this chapter

Deployment types

The User Management Interface software supports network access directly to the server, network access through a load balancer, and local access. After you run the installation, you run a script at the command line, in which you state the type of deployment you want to use. For more information, see ***Choosing the deployment type*** (on page 20).

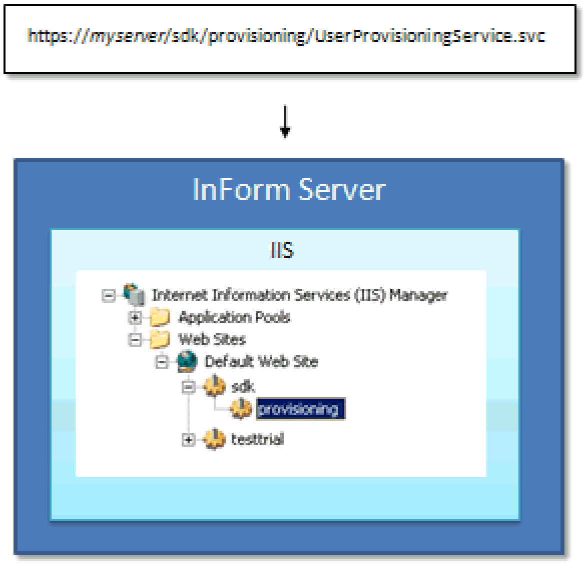
Deployment types			
Deployment name	HTTP / HTTPS	Authentication?	URL / SOAP study matching?
Default	HTTPS	Yes	No
Load balanced	HTTP *	Yes	Yes
LAN access	HTTP	No	No
* Represents the data sent by the load balancer to the API.			

Default deployment

Default deployment is configured automatically by the product installer and allows clients to securely access the server through the top-level virtual directory. An SSL certificate must be installed for the provisioning virtual directory and the request must include a valid user name/password used to authenticate the requests. When deployed in this manner, the API is not exposed beneath the study URL and is, instead, exposed as a top-level directory and, therefore, no matching of the study name in the URL against the SOAP request occurs.

Example client URL:

```
https://myserver/sdk/provisioning/UserProvisioningService.svc
```

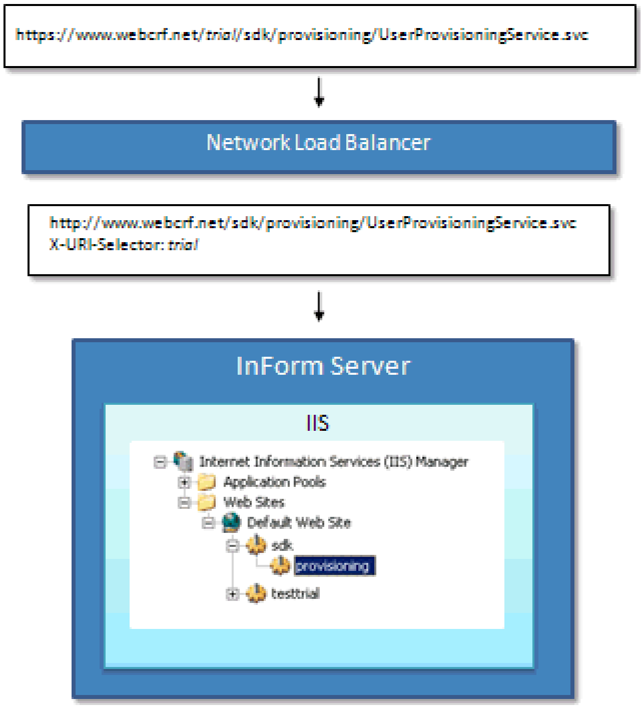

https://myserver/sdk/provisioning/UserProvisioningService.svc
InForm Server
IIS
Internet Information Services (IIS) Manager
Application Pools
Web Sites
Default Web Site
sdk
provisioning
testtrial

Load balanced deployment

Load balanced deployment is configured through modifications to the web.config file and is intended for use when exposing the API beneath the InForm study URL through the use of a network load balancer. When deployed in this manner, the load balancer is responsible for SSL decryption (no certificate is installed for the provisioning virtual directory) as well as altering the request URL to remove the study name from the request. Requests must include a valid user name/password as well as the HTTP X-URI-Selector header containing the name of the study that was originally specified in the requested URL. For more information, see ***Choosing the deployment type*** (on page 20).

Example client URL:

```
https://myserver/trialX/sdk/provisioning/UserProvisioningService.svc
```


https://www.webcrf.net/trial/sdk/provisioning/UserProvisioningService.svc
Network Load Balancer
http://www.webcrf.net/sdk/provisioning/UserProvisioningService.svc
X-URI-Selector: trial
InForm Server
IIS
Internet Information Services (IIS) Manager
Application Pools
Web Sites
Default Web Site
sdk
provisioning
testtrial

LAN access deployment

LAN access deployment allows clients on the same network-secured LAN to access the API without security. As with default security, when deployed in this manner, the API is not exposed beneath the study URL and, therefore, no study name matching is performed.

Example client URL:

```
http://myserver/sdk/provisioning/UserProvisioningService.svc
```

Choosing the deployment type

The User Management Interface software comes with:

- Three variations of the web.config file to support the different configurations that are used to deploy provisioning.
- The utility ProvisioningWebConfigFileSelector.cmd, which you use to set the appropriate configuration file for your configuration type.

To select the deployment type:

1 Open a Command Prompt window and change to the <provisioning_install_location>\Provisioning\ directory.

2 Issue the following command:

```
ProvisioningWebConfigFileSelector.cmd CONFIGURATION
```

where

CONFIGURATION specifies the deployment type.

For example:

```
ProvisioningWebConfigFileSelector.cmd F5
```

The utility replaces the InForm SDK\Provisioning\web.config file with one of the following files.

Deployment configuration options			
Deployment type	Deployment type configuration option	Config file	Details
Default	SECURE	Web_secure.config	• Secure configuration for client requests sent over the Internet. • Supports HTTPS transport using SOAP 1.2. • Requires a Username Token in the SOAP Security header for providing the user name and password credentials to authenticate a request.

Deployment configuration options			
Deployment type	Deployment type configuration option	Config file	Details
Load balanced	F5	Web_secure_F5.config	• Secure configuration for client requests sent over the Internet using a network load balancer. • Supports HTTPS into the network load balancer and HTTP out of it using SOAP 1.2. • Requires a Username Token in the SOAP Security header for providing the user name and password credentials to authenticate a request. • Verifies that the study name in the URL sent to the network load balancer matches the study name parameter in the SOAP request.
LAN access	UNSECURE	Web_nosecurity.config	• Unsecure configuration for client requests sent over a LAN from behind a firewall. • Supports HTTP transport and SOAP 1.2. • Does not perform request authentication.

Study request restrictions

When deployed in the load balanced configuration, the requestor is prevented from passing in alternate study names (study names not matching that in the URL) as part of the request. For example:

```
URL: https://myserver/trial1/sdk/provisioning/UserProvisioningService.svc
Study in SOAP request: trial2
```

To support enforcement of matching the study in the request to the name of the study in the URL, a custom HTTP header is supported by the User Management Interface software. The X-URI-Selector can contain the name of the study that was specified in the original request to the load balancer. For example:

```
X-URI-Selector: trial1
```

When deployed in the load balanced configuration, this header element must be present and match the study name specified in the SOAP request. For more information, see ***SOAP header*** (on page 27).

CHAPTER 4

Programming model and conventions

In this chapter

How requests are processed

Changes that you specify in a single call occur as part of an atomic unit and are applied immediately on successful completion of the call. If one change contains an error, none of the changes are applied. For example:

- If you attempt to add five users, and you specify a disallowed character for one user name, none of the users are added.
- If you attempt to remove four users from a group, but one of the users does not exist in the study, none of the users are removed from the group.

Requests are subject to the following constraints:

- **Processing time**—The maximum allowed time for a single request is five minutes. Processing that exceeds the maximum time results in an error.
- **Size of requests—**Requests must be smaller than 5 MB.
- **Case-sensitivity**—The User Management Interface software requires the same case-sensitivity rule as the InForm application. Object references by name are case-sensitive. Objects whose names contain different cases (for example, "cra" and "CRA") are different objects.
- **Sequence of operations**—The caller is responsible for sequencing operations as required for semantic correctness (for example, users must exist before you can add them to a rights group).

Before the request is processed, the following operations are performed:

- The server logs the request (operation and user name only) to the event log for historical analysis.
- If you use a load balanced deployment method, the study name parameter is validated against the study name in the URL. If the two do not match, an error is returned.

  For more information, see ***Load balanced deployment*** (on page 19).
- The user name and password are extracted from the SOAP header and are validated against the InForm database. Only requests with valid credentials for active accounts are accepted and processed.

  Multiple repeat requests with invalid passwords will disable the account and require administrative action according to existing InForm business logic. The allowed number of failed log-on attempts is set in the InForm application on the System Configuration page.
- Parameters are validated. If the request contains invalid content, an error is returned. For example, schema validation and InForm character restrictions are enforced.

The API uses a subset of the InForm MedML schema to describe provisioning data.

Writing requests

Each request is enclosed in a SOAP envelope that begins with a SOAP header, followed by a Body statement that includes the operation and field values.

For examples of Body statements, see the individual *Operations* (on page 33).

Note: The User Management Interface documentation ISO includes a ZIP file that contains complete XML file examples.

WSDL

The API definition is included in the WSDL.

The WSDL and associated schemas are included in the InForm SDK\Provisioning\wsdl directory of the installation directory.

Because WSDL contain references to the server on which they were generated, you must update the server name in the WSDL before the WSDL can be used in your program.

Update the server name manually.

The following example of WSDL code shows in **bold** text the areas you must update.

```
<wsdl:service name="UserProvisioningService">
   <wsdl:port name="UserProvisioningService"
   binding="tns:UserProvisioningService">
      <soap12:address
      location="https://machineName/sdk/provisioning/UserProvisioningService.s
      vc" />
      <wsa10:EndpointReference>
         <wsa10:Address>https://machineName/sdk/provisioning/UserProvisioningS
         ervice.svc</wsa10:Address>
      </wsa10:EndpointReference>
   </wsdl:port>
</wsdl:service>
```

SOAP header

The SOAP header contains information about the service, the message, and the sender. It must include all of the following fields.

> **Note: The requirement for requests to contain the wsu:Timestamp, wsse:Username, and wsse:Password elements are defined in the WSDL by ws-policy.**
>
> **If your web service toolkit does not support ws-policy, you must manually add these elements to the web service request.**

Field	Description	Standard
wsa:Action	SOAP action for the operation. Include this field as part of the summary for each soap operation.	WS-Addressing
wsa:MessageID	Unique ID for the message.	WS-Addressing
wsa:ReplyTo	Use the value that is shown in the ***example*** (on page 28).	WS-Addressing
wsa:To	Full URL of the service. **Note:** The full URL is different for each customer.	WS-Addressing
wsu:Timestamp	Time the message was created and when it expires. Oracle recommends a 5-minute time window. The software allows for five minutes of clock drift between servers. This value can be modified in the web.config file by setting the MaxClockSkew value. For more information, see ***Increasing the allowable time difference between client and server*** (on page 29).	SOAP Message Security 1.0
wsse:Username	User name of the integration user in the InForm study.	SOAP Message Security 1.0
wsse:Password	Password of the integration user in the InForm study.	SOAP Message Security 1.0

Example: SOAP header

This example shows a header for a call to the GetUserNames operation.

Note: The **wsa:Action** value, shown in **bold** text in the example, is different for each API operation.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=http://www.w3.org/2003/05/soap-envelope
xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <SOAP-ENV:Header>
      <wsa:Action
      xmlns:wsa="http://www.w3.org/2005/08/addressing">http://www.phaseforward.c
      om/InForm/2009/01/UserProvisioning/GetUserNames</wsa:Action>
      <wsa:MessageID
      xmlns:wsa="http://www.w3.org/2005/08/addressing">uuid:c266299a-721d-473e-
      b548-7ca1706eeff0</wsa:MessageID>
      <wsa:ReplyTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
         <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address
         >
      </wsa:ReplyTo>
      <wsa:To
      xmlns:wsa="http://www.w3.org/2005/08/addressing">https://myserver/PFST45/s
      dk/provisioning/UserProvisioningService.svc</wsa:To>
   <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
   200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
   open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsu:Timestamp>
         <wsu:Created>2009-08-19T19:47:06Z</wsu:Created>
         <wsu:Expires>2009-08-19T19:52:06Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:UsernameToken>
         <wsse:Username>testuser</wsse:Username>
         <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
         200401-wss-username-token-profile-
         1.0#PasswordText">testpassword</wsse:Password>
      </wsse:UsernameToken>
   </wsse:Security>
</SOAP-ENV:Header>
```

Increasing the allowable time difference between client and server

The value of MaxClockSkew, which is the allowable time difference between the client machine clock and the InForm server clock, can be modified in the web.config file for the User Management Interface.

If the time difference between the client machine clock and the InForm server machine clock is greater than the value specified by the MaxClockSkew setting, the WCF rejects the SOAP message with a security error, and the operation fails.

The default clock skew is five minutes.

To change the allowable time difference:

1 Using an editor of your choice, open the User Management Interface web.config file, located at

<installation_directory>\Provisioning\web.config

2 Locate the following line, which in the default installation, includes opening comment characters (<!--) and closing comment characters (-->):

```
<!--
   <add key="InForm.SDK.Provisioning.Settings.Web.MaxClockSkew"
   value="00:10:00"/>
-->
```

3 Delete the opening comment characters **<!--** and the closing comment characters **-->**.

4 Set the value of the clock skew. The format is hours:minutes:seconds.

For example:

Time value	Setting
10 minutes	value="00:10:00"
One hour	value="01:00:00"
90 minutes	value="01:30:00"

5 Save the file.

Identifier sets

The following User Management Interface operations update or insert data into the InForm database:

- *PutProvisioningDataAndGetIdentifierSets* (on page 49).
- RemoveUsersFromGroups.
- *RemoveAllUsersFromGroups* (on page 52).

These operations return identifier set information for the items that have been updated or inserted. Identifier sets provide more information than that provided by MedML, and can help clients maintain metadata about particular entities in their own systems.

Identifier sets contain the following components:

IdentifierSetList component	Information
HasStaleIdentifierSets	Values: true, false. Indicates whether any of the identifier sets have been returned the STALE identifier with a value of **true**.
IdentifierSet	Contains information about one site, user, or group on which the operation acted.

Each identifier set contains information about a single item.

IdentifierSet identifier	Information
DBUID	Unique key in the InForm database schema.
GUID	GIUD of the entity.
REVISION MAXHISTORICALORDER	Can be stored on the client side and used to determine the age or staleness of copies of entity data by comparing the values with more recent numbers. For example, an increase in the MAXHISTORICALORDER value for an entity since the last time the client retrieved that entity, the data might be stale and should be refreshed.
STALE	Values: true, false. Client can choose to retry stale identifier sets by sending them back to the ***GetIdentifierSets operation*** (on page 36) until the revision history or max historical order increase.

Example: Identifier sets

The example shows an identifier set list with identifier sets for a site and a user.

```
<IdentifierSetList>
   <HasStaleIdentifierSets>false</HasStaleIdentifierSets>
   <IdentifierSet>
        <Name>Hospital ABC</Name>
        <TYPE>SITE</TYPE>
        <DBUID>22421</DBUID>
        <GUID>{B9FFFC0A-2C2D-49CB-B1D2-A92D299159F2}</GUID>
        <REVISION>333222253222</REVISION>
        <MAXHISTORICALORDER>100303</MAXHISTORICALORDER>
        <STALE>false</STALE>
   </IdentifierSet>
   <IdentifierSet>
        <Name>bsmith</Name>
        <TYPE>USER</TYPE>
        <DBUID>3421</DBUID>
        <GUID>{20BF745D-B982-41EE-8DA9-9E4ACF49C9A8}</GUID>
        <REVISION>33322225777</REVISION>
        <MAXHISTORICALORDER>100310</MAXHISTORICALORDER>
        <STALE>false</STALE>
   </IdentifierSet>
```

Best practices

Choosing whether to get identifier sets when putting data

Calls to each of the following operations have the same effect on the InForm study database:

- PutProvisioningData
- *PutProvisioningDataAndGetIdentifierSets* (on page 49)

PutProvisioningDataAndGetIdentifierSets returns the identifier sets associated with the changed entities in the database, and is significantly slower than PutProvisioningData.

Use PutProvisioningDataAndGetIdentifierSets if the returned identifier sets are used within the client.

Using the User Management Interface with the InForm 4.6 or InForm 5.0 software

Requests for put data operations in the User Management Interface software include the MedML elements <USER> and <SITE>, which contain one or more of the following fields:

Field	Used in MedML element
PRODUCTLOCALE	USER
STUDYLOCALE	USER, SITE
USERNAMEORDER	SITE

These fields accommodate internationalization features in the InForm 5.0 and subsequent software releases. If you use the User Management Interface software with InForm 4.6 studies, these field values are ignored.

Oracle recommends that you include these fields in the USER and SITE elements in any requests for User Management Interface operations, regardless of whether you are using the User Management Interface software with the InForm 4.6 or InForm 5.0 releases (and subsequent supported releases in the InForm 5.x and 6.x release streams.

Using the User Management Interface with the InForm 5.5 and later software

If you are using the InForm 5.5 SP0b or later software release, new users that you add with the User Management Interface must have an associated rights group in order to be able to log in to the InForm application.

CHAPTER 5

Operations

In this chapter

AddUsersToSite

Use	Associates a list of users to a specified site group.
Inputs	• Trial name. • Site name. • List of user names.
Outputs	None on success, SOAP fault on failure.
Side effects	Users are associated to the specified site group. **Note:** If the InForm system user is part of the list of users, it will not be added. A message is logged to the server's event log, stating that the system user was skipped. No error is reported. To add the system user to a site, use the InForm user interface.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/AddUsersToSite

Example: AddUsersToSite

Example Request—This example adds six users to the site.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <AddUsersToSite>
      <TrialName>pfst45</TrialName>
      <SiteName>(01) Massachusetts General Hospital</SiteName>
      <UserNames>
         <string>ActiveNotDeletedUser1-0653e706</string>
         <string>afarmington</string>
         <string>ajones</string>
         <string>aortega</string>
         <string>astone</string>
         <string>AuthUser-0038b973</string>
        </UserNames><Extensions/>
   </AddUsersToSite>
</s:Body>
```

CookMedML

Use	Cooks any MedML document that is accepted by the interface. InForm provisioning will cook the MedML as is.
Inputs	• Study name. • MedML document.
Outputs	Empty response if success; otherwise, SOAP fault.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/CookMedML

Example: CookMedML

Example Request

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <CookMedML>
      <TrialName>PFST45</TrialName>
      <MedMLDoc><MEDMLDATA xmlns="PhaseForward-MedML-Inform4"><SYSCONFIG
      CONFIGNAME="QueryMaxLength" TYPE="0" VALUE="360"></SYSCONFIG><SYSCONFIG
      CONFIGNAME="DaysPasswordExpiration" TYPE="0"
      VALUE="60"></SYSCONFIG></MEDMLDATA></MedMLDoc>
      <Extensions/>
   </CookMedML>
</s:Body>
```

Example Response—Empty response if successful, otherwise, SOAP fault.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <CookMedMLResponse>
   </CookMedMLResponse>
</s:Body>
```

GetIdentifierSetList

Use	Returns a fresh IdentifierSetList for IdentifierSets of type USER, SITE, RIGHTSGROUP, QUERYGROUP, SIGNATUREGROUP, and REPORTINGGROUP without performing any create or update operations.
Inputs	• Study name. • IdentifierSetList containing an IdentifierSet for each user, site, or supported group type about which to retrieve updated information. Specify the NAME and TYPE attributes. Any other specified attributes are ignored. **Note:** Although the other nodes (HasStaleIdentifierSets, DBUID, GUID, REVISION, MAXHISTORICALORDER and STALE) are required, their values are ignored. Supported group types are RIGHTSGROUP, QUERYGROUP, SIGNATUREGROUP, and REPORTINGGROUP.
Outputs	For each site, group, or user that is specified by an identifier set in the request, an identifier set is returned if the specified entity exists in the study database.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/GetIdentifierSetList

For more information, see ***Identifier sets*** (on page 30).

Example: GetIdentifierSetList

Example request—This example requests identifier sets for a site, user, and query group.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetIdentifierSetList>
      <TrialName>pfst45</TrialName>
      <IdentifierSetList>
         <IdentifierSet>
            <Name>(02) Cleveland Clinic</Name>
            <TYPE>SITE</TYPE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>mcarlson</Name>
            <TYPE>USER</TYPE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>CRA Query</Name>
            <TYPE>QUERYGROUP</TYPE>
         </IdentifierSet>
      </IdentifierSetList>
   </GetIdentifierSetList>
</s:Body>
```

Example response

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <GetIdentifierSetListResponse>
      <IdentifierSetList>
         <HasStaleIdentifierSets>false</HasStaleIdentifierSets>
         <IdentifierSet>
            <Name>mcarlson</Name>
            <TYPE>USER</TYPE>
            <DBUID>9568</DBUID>
            <GUID>{F453A59D-B098-4277-B188-E1A410605034}</GUID>
            <REVISION>275587031321086</REVISION>
            <MAXHISTORICALORDER>18723</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>(02) Cleveland Clinic</Name>
            <TYPE>SITE</TYPE>
            <DBUID>10334</DBUID>
            <GUID>{9005B081-2469-4219-BDC1-D45790991A5F}</GUID>
            <REVISION>275587048243086</REVISION>
            <MAXHISTORICALORDER>5362</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>CRA Query</Name>
            <TYPE>QUERYGROUP</TYPE>
            <DBUID>10558</DBUID>
            <GUID>{3CC89144-A200-4E10-8AE5-33ACE70F84B4}</GUID>
            <REVISION>275587051556086</REVISION>
            <MAXHISTORICALORDER>4824</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
      </IdentifierSetList>
   </GetIdentifierSetListResponse>
</s:Body>
```

GetInFormVersion

Use	Returns the InForm version.
Inputs	Study name.
Outputs	The InForm major and minor version. For example: 4.6, 5.0, 5.5, 6.0, 6.1, and so on.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/ GetInFormVersion

Example: GetInFormVersion

Example Request

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetInFormVersion>
      <TrialName>PFST45</TrialName>
      <Extensions/>
   </GetInFormVersion>
</s:Body>
```

Example Response—The response contains a string value representing the InForm major and minor version.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <GetInFormVersionResponse>
      <GetInFormVersionResult>4.6</GetInFormVersionResult>
   </GetInFormVersionResponse>
</s:Body>
```

GetLatestStudyVersion

Use	Returns the latest study version in the study, excluding InForm system study versions.
Inputs	Study name.
Outputs	The latest study version in the study. This is taken from the MedML STUDYVERSION element.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/ GetLatestStudyVersion

Example: GetLatestStudyVersion

Example Request

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   < GetLatestStudyVersion>
      <TrialName>PFST45</TrialName>
      <Extensions/>
   </ GetLatestStudyVersion>
</s:Body>
```

Example Response—The response contains MedML <STUDYVERSION> elements in a LatestStudyVersion element.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   < GetLatestStudyVersionResponse>
      < LatestStudyVersion>
         <STUDYVERSION STUDYNAME="PF-ST Sample Trial"
         VERSIONDESCRIPTION="StudyVersion1"REVISION="275587061728086"
         GUID="{96D6B3F4-4E54-48B4-A625-15987933D6FA}" />
      </ LatestStudyVersion>
   </ GetLatestStudyVersionResponse>
</s:Body>
```

GetProvisioningVersion

Use	Returns the version of the User Management Interface software.
Inputs	None.
Outputs	String that contains the version of the User Management Interface software.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/GetProvisioningVersion

Example: GetProvisioningVersion

Example request

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetProvisioningVersion>
      <TrialName>pfst45</TrialName>
   </GetProvisioningVersion>
</s:Body>
```

Example response—In this example, the operation returns the version of the User Management Interface software in the value for the ProvisioningVersion element.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <GetProvisioningVersionResponse>
      <ProvisioningVersion>1.0.83</ProvisioningVersion>
   </GetProvisioningVersionResponse>
</s:Body>
```

GetStudyVersions

Use	Returns the list of study versions in the trial in the order they were created, excluding InForm system study versions.
Inputs	Study name.
Outputs	List of study versions in the study (through the MedML STUDYVERSION element). Study versions are returned in the order they were created in the study.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/GetStudyVersions

Example: GetStudyVersions

Example request

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetStudyVersions>
      <TrialName>PFST45</TrialName>
      <Extensions/>
   </GetStudyVersions>
</s:Body>
```

Example response—The response contains MedML <STUDYVERSION> elements in a StudyVersion List

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <GetStudyVersionsResponse>
      <StudyVersionList>
         <STUDYVERSION STUDYNAME="PF-ST Sample Trial"
         VERSIONDESCRIPTION="StudyVersion1"REVISION="275587061728086"
         GUID="{96D6B3F4-4E54-48B4-A625-15987933D6FA}" />
      </StudyVersionList>
   </GetStudyVersionsResponse>
</s:Body>
```

GetUserDetails

Use	Returns all attributes for users whose user names you specify.
Inputs	• Study name. • Array of user names.
Outputs	USERs that match the specified user names. **Note:** This operation returns a MedML response, not IdentifierSets.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/GetUserDetails

Example: GetUserDetails

Example request—This example lists two users.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetUserDetails>
      <TrialName>pfst45</TrialName>
         <UserNames>
            <string>ActiveNotDeletedUser1-0653e706</string>
            <string>afarmington</string>
            <string>AuthUser-0038b973</string>
         </UserNames>
      <Extensions/>
   </GetUserDetails>
</s:Body>
```

Example response—The response shows a user list with details for each user.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <GetUserDetailsResponse>
      <UserList>
         <USER USERNAME="ActiveNotDeletedUser1-0653e706" USERTYPE="SYSTEM"
         GUID="{F6767564-28D4-42B1-B864-143FFBE4B9C6}" REVISION="287067082438097"
         ACTIVESTATE="TRUE" DELETESTATE="FALSE" USERMUSTRESETPASSWORD="FALSE"
         USERDATEFORMAT="MONTH_DAY_YEAR" FIRSTNAME="First1" LASTNAME="Last1"
         DISPLAYNAME="Display1" DESCRIPTION="desc" TITLE="title" BEEPER="beeper"
         HOMESCREENURL="url" ADDRESS="Address1" ADDRESS2="address2" CITY="City"
         STATE="state" PROVINCE="state" ZIPCODE="zip" POSTCODE="zip"
         COUNTRY="country" PHONE="1234" ALTPHONE="12345" FAX="faxUpdate"
         EMAIL="update@a.com" />
         <USER USERNAME="afarmington" USERTYPE="SITE" GUID="{1E528D19-2235-46DF-
         AEC1-2F1ED6FA4EB8}" REVISION="275587035852086" ACTIVESTATE="TRUE"
         DELETESTATE="FALSE" USERMUSTRESETPASSWORD="FALSE"
         USERDATEFORMAT="MONTH_DAY_YEAR" FIRSTNAME="Abby" LASTNAME="Farmington"
         DISPLAYNAME="Abby Farmington" HOMESCREENURL="./Custom/HomeDefault.html"
         PHONE="(555) 555-1212" FAX="(555) 555-1212" EMAIL="email@domain.test" />
         <USER USERNAME="AuthUser-0038b973" USERTYPE="SYSTEM" GUID="{ACA36CA1-
         47D8-482D-8AAF-DF1FF8FA3E82}" REVISION="287068464402074"
         ACTIVESTATE="TRUE" DELETESTATE="FALSE" USERMUSTRESETPASSWORD="FALSE"
         USERDATEFORMAT="MONTH_DAY_YEAR" FIRSTNAME="First1" LASTNAME="Last1"
         DISPLAYNAME="Display1" DESCRIPTION="desc-Updated" TITLE="title"
         BEEPER="beeper" HOMESCREENURL="url" ADDRESS="Address1"
         ADDRESS2="address2" CITY="City" STATE="state" PROVINCE="state"
         ZIPCODE="zip" POSTCODE="zip" COUNTRY="country" PHONE="1234"
         ALTPHONE="12345" FAX="faxUpdate" EMAIL="update@a.com" />
      </UserList>
   </GetUserDetailsResponse>
</s:Body>
```

GetUserNames

Use	Returns the list of user names in the study.
Inputs	Study name and optional filter. A percent sign (%) indicates a wildcard. For example, "S%" returns any user name that starts with S or s. Differences in case are ignored. If the study contains user names that differ only in letter case, all matching names are retrieved.
Outputs	List of user names in the study that match the optional filter.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/GetUserNames

Example: GetUserNames

Example Request—This example retrieves all the user names in study **testtrial**.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetUserNames>
      <TrialName>testtrial</TrialName>
      <Filter/>
      <Extensions/>
   </GetUserNames>
</s:Body>
```

Example Response—This example shows six of the user names that were returned. The remaining user names have been removed from the example for brevity.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <GetUserNamesResponse>
      <UserNameList>
         <USER USERNAME="afarmington" USERTYPE="SITE" GUID="{1E528D19-2235-46DF-
         AEC1-2F1ED6FA4EB8}" REVISION="275587035852086" />
         <USER USERNAME="ajones" USERTYPE="SITE" GUID="{FB73AEB7-DD22-4C7B-A024-
         76C845330801}" REVISION="275587033134086" />
         <USER USERNAME="aortega" USERTYPE="SITE" GUID="{B3925738-9FDF-4676-BC0E-
         1F2162A418DB}" REVISION="275587040509086" />
         <USER USERNAME="awarren" USERTYPE="SITE" GUID="{B3D13070-68D1-4D85-9666-
         65DC0230FA76}" REVISION="275587027649086" />
         <USER USERNAME="cmendoza" USERTYPE="SITE" GUID="{5BB95E51-9857-45E2-
         ABBD-2FFB832706AC}" REVISION="275587039587086" />
         <USER USERNAME="cmitchell" USERTYPE="SITE" GUID="{0FCBFCC7-3134-48C5-
         8BC5-CF1D22CB6DDD}" REVISION="275587032212086" />
         <!-- additional names not shown in example -->
      </UserNameList>
   </GetUserNamesResponse>
</s:Body>
```

GetUserNamesByType

Use	Returns a list of user names in the study.
Inputs	• Study name. • Optional filter. A percent sign (%) indicates a wildcard. For example, "S%" returns any user name that starts with S or s. Differences in case are ignored. If the study contains user names that differ only in letter case, all matching names are retrieved. • InFormUserType (SITE or SYSTEM or SPONSOR or INTEGRATION or SUPPORT).
Outputs	List of user names in the study, represented as MedML <USER> elements, that match the optional filter. **Note:** This operation returns a MedML response, not IdentifierSets.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/GetUserNamesByType

Example: GetUserNamesByType

Example Request—This example retrieves a list of SITE users in the study.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetUserNamesByType>
      <TrialName>pfst45</TrialName>
      <Filter>%%</Filter><TypeFilter>SITE</TypeFilter>
   <Extensions/></GetUserNamesByType>
</s:Body>
```

Example Response—The Body of the response includes a list of the SITE users.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <GetUserNamesByTypeResponse>
      <UserNameList>
         <USER USERNAME="afarmington" USERTYPE="SITE" GUID="{1E528D19-2235-46DF-
         AEC1-2F1ED6FA4EB8}" REVISION="275587035852086" />
         <USER USERNAME="ajones" USERTYPE="SITE" GUID="{FB73AEB7-DD22-4C7B-A024-
         76C845330801}" REVISION="275587033134086" />
         <USER USERNAME="aortega" USERTYPE="SITE" GUID="{B3925738-9FDF-4676-BC0E-
         1F2162A418DB}" REVISION="275587040509086" />
         <USER USERNAME="awarren" USERTYPE="SITE" GUID="{B3D13070-68D1-4D85-9666-
         65DC0230FA76}" REVISION="275587027649086" />
         <USER USERNAME="cmendoza" USERTYPE="SITE" GUID="{5BB95E51-9857-45E2-
         ABBD-2FFB832706AC}" REVISION="275587039587086" />
         <USER USERNAME="cmitchell" USERTYPE="SITE" GUID="{0FCBFCC7-3134-48C5-
         8BC5-CF1D22CB6DDD}" REVISION="275587032212086" />
      </UserNameList>
   </GetUserNamesByTypeResponse>
</s:Body>
```

GetUserSites

Use	Returns a list of sites to which the user belongs.
Inputs	• Study name. • User name.
Outputs	SITEs that the user belongs to. **Note:** This operation returns a MedML response, not IdentifierSets.
Side effects	None.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/GetUserSites

Example: GetUserSites

Example request—This example requests a list of sites that user ajones belongs to for trial pfst45.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <GetUserSites>
      <TrialName>pfst45</TrialName>
      <UserName>ajones</UserName>
      <Extensions/>
   </GetUserSites>
</s:Body>
```

Example response—The response contains a list of sites.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <GetUserSitesResponse>
      <SiteList>
         <SITE NAME="(01) Massachusetts General Hospital" MNEMONIC="01"
         GUID="{7BB62C72-F669-4CF5-8E23-0C137326952C}" REVISION="275587047775086"
         STARTDATE="4/1/2005" ENDDATE="12/30/1899"
         SITEDATEFORMAT="MONTH_DAY_YEAR" SITESERVER="RDINFAPP116" TIMEZONE="(GMT-
         05:00) Eastern Time (US &amp; Canada)" ADDRESS="1 Main Street"
         CITY="Boston" STATE="MA" PROVINCE="MA" COUNTRY="USA" PHONE="(555) 555-
         1212" />
         <SITE NAME="(02) Cleveland Clinic" MNEMONIC="02" GUID="{9005B081-2469-
         4219-BDC1-D45790991A5F}" REVISION="275587048243086" STARTDATE="4/1/2005"
         ENDDATE="12/30/1899" SITEDATEFORMAT="MONTH_DAY_YEAR"
         SITESERVER="RDINFAPP116" TIMEZONE="(GMT-06:00) Central Time (US &amp;
         Canada)" ADDRESS="202 Walnut Street" CITY="Cleveland" STATE="OH"
         PROVINCE="OH" COUNTRY="USA" PHONE="(555) 555-1212" />
         <SITE NAME="(05) University of California Hospital" MNEMONIC="05"
         GUID="{8AB02ADB-571E-48A7-8AD7-0D7881527EEA}" REVISION="275587049478086"
         STARTDATE="4/1/2005" ENDDATE="12/30/1899"
         SITEDATEFORMAT="MONTH_DAY_YEAR" SITESERVER="RDINFAPP116" TIMEZONE="(GMT-
         08:00) Pacific Time (US &amp; Canada); Tijuana" ADDRESS="62 Hospital
         Way" CITY="Irvine" STATE="CA" PROVINCE="CA" COUNTRY="USA" PHONE="(555)
         555-1212" />
      </SiteList>
   </GetUserSitesResponse>
</s:Body>
```

PutProvisioningData

Calls to PutProvisioningData are processed serially in the order they are received by the server. The processing of each call will not exceed five minutes.

Use	Creates or updates a set of provisioning data for one or more users and sites.
Inputs	• Study name. • One or more of: User, Site, RightsGroup, SiteGroup, StudyVersionSite, QueryGroup, SignatureGroup, ReportingGroup.
Outputs	None on success, SOAP fault on failure.
Side effects	Items are created or updated based on input parameters.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/ PutProvisioningData

Note: If you are using the InForm 5.5 SP0b or later software release, new users that you add with the User Management Interface must have an associated rights group in order to be able to log in to the InForm application.

Example: PutProvisioningData

Example request 1—This example activates two users.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <PutProvisioningData>
      <TrialName>testtrial</TrialName>
      <MedML xmlns:pfupml="PhaseForward-MedML-Inform4">
         <USER USERNAME="SiteUser2" ACTIVESTATE="TRUE">
         <USER USERNAME="SponsorUser2" ACTIVESTATE="TRUE">
      </MedML>
      <Extensions/>
   </PutProvisioningData>
</s:Body>
```

Example request 2—This example adds two users in the inactive state.

```
<s:Body>
   <PutProvisioningData
   xmlns="http://www.phaseforward.com/InForm/2009/01/UserProvisioning">
      <TrialName>testtrial</TrialName>
      <MedML>
         <USER USERNAME="SiteUser2" USERTYPE="SITE" FIRSTNAME="First1"
         LASTNAME="Last1" DISPLAYNAME="SiteDisplay2" DESCRIPTION="desc"
         TITLE="title" ADDRESS="Address1" ADDRESS2="address2" CITY="City"
         STATE="state" ZIPCODE="zip" COUNTRY="country" PHONE="1234"
            ALTPHONE="12345" FAX="faxUpdate" EMAIL="update@a.com" BEEPER="beeper"
            HOMESCREENURL="url" ACTIVESTATE="FALSE" DELETESTATE="FALSE"
            USERMUSTRESETPASSWORD="FALSE" PASSWORD="ABCDEF">
         <USER USERNAME="SponsorUser2" USERTYPE="SPONSOR" FIRSTNAME="First1"
         LASTNAME="Last1" DISPLAYNAME="SponsorDisplay2" DESCRIPTION="desc"
         TITLE="title" ADDRESS="Address1" ADDRESS2="address2" CITY="City"
         STATE="state" ZIPCODE="zip" COUNTRY="country"
            PHONE="1234" ALTPHONE="12345" FAX="faxUpdate" EMAIL="update@a.com"
            BEEPER="beeper" HOMESCREENURL="url" ACTIVESTATE="FALSE"
            DELETESTATE="FALSE" USERMUSTRESETPASSWORD="FALSE" PASSWORD="ABCDEF">
      </MedML>
      <Extensions/>
   </PutProvisioningData>
</s:Body>
```

PutProvisioningDataAndGetIdentifierSets

Use	Creates or updates a set of provisioning data for one or more users and sites. Returns identifier set information for all sites, groups, and users in the MedML elements.
Inputs	• Trial name. • One or more of: User, Site, RightsGroup, SiteGroup, StudyVersionSite, QueryGroup, SignatureGroup, ReportingGroup.
Outputs	Identifier set list, containing an identifier set for each item that has been modified. Items include: users, sites, rights groups, query groups, signature groups, reporting groups.
Side effects	Items are created or updated based on input parameters. **Note: PutProvisioningDataAndGetIdentifierSets** has the same effect on the InForm study database as **PutProvisioningData**. Use **PutProvisioningDataAndGetIdentiferSets** if the returned identifier sets are used within the client, because the method is significantly slower than **PutProvisioningData**.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/PutProvisioningDataAndGetIdentiferSets

Note: If you are using the InForm 5.5 SP0b or later software release, new users that you add with the User Management Interface must have an associated rights group in order to be able to log in to the InForm application.

Example: PutProvisioningDataAndGetIdentifierSets

Example request—This example adds two sites to a study.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <PutProvisioningDataAndGetIdentifierSetList>
   <TrialName>pfst45</TrialName>
      <MedML xmlns:pfupml="PhaseForward-MedML-Inform4">
         <SITE NAME="UT-Site-20b14482-78ec-46f5-8e05-d3fca70e3111"
         MNEMONIC="20b1448278ec46f58e05d3fca70e3111" ADDRESS="7700 McGill
         Boulevard" ADDRESS2="address2" CITY="Montreal" STATE="Quebec"
         ZIPCODE="12345" COUNTRY="CANADA" PHONE="(555) 555-1212" ALTPHONE="12345"
         FAX="fax" EMAIL="@a.com" BEEPER="beeper" TIMEZONE="(GMT-05:00) Eastern
         Time (US &amp;amp; Canada)" STARTDATE="4/1/2005" ENDDATE="4/1/2010"
         SITEDATEFORMAT="MONTH_DAY_YEAR">
         <STUDYVERSIONSITE SITEMNEMONIC="20b1448278ec46f58e05d3fca70e3111"
         VERSIONDESCRIPTION="StudyVersion1" ACCEPTDATE="4/1/2005" REASON="Study
         Version Site Update">

         <SITE NAME="UT-Site-dfe4ddcf-73e5-4c47-b523-50fc47cf9c54"
         MNEMONIC="dfe4ddcf73e54c47b52350fc47cf9c54" ADDRESS="7700 McGill
         Boulevard" ADDRESS2="address2" CITY="Montreal" STATE="Quebec"
         ZIPCODE="12345" COUNTRY="CANADA" PHONE="(555) 555-1212" ALTPHONE="12345"
         FAX="fax" EMAIL="@a.com" BEEPER="beeper" TIMEZONE="(GMT-05:00) Eastern
         Time (US &amp;amp; Canada)" STARTDATE="4/1/2005" ENDDATE="4/1/2010"
         SITEDATEFORMAT="MONTH_DAY_YEAR">
         <STUDYVERSIONSITE SITEMNEMONIC="dfe4ddcf73e54c47b52350fc47cf9c54"
         VERSIONDESCRIPTION="StudyVersion1" ACCEPTDATE="4/1/2005" REASON="Study
         Version Site Update">
        </MedML>
      <Extensions/>
   </PutProvisioningDataAndGetIdentifierSetList>
</s:Body>
```

Example response—The response contains an identifier set for each site that was added.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
   <s:Body>
      <PutProvisioningDataAndGetIdentifierSetListResponse>
         <IdentifierSetList>
            <HasStaleIdentifierSets>false</HasStaleIdentifierSets>
            <IdentifierSet>
               <Name>UT-Site-20b14482-78ec-46f5-8e05-d3fca70e3111</Name>
               <TYPE>SITE</TYPE><DBUID>56142</DBUID>
               <GUID>{A9BAC620-BA31-47A2-A8B6-9504F2900D84}</GUID>
               <REVISION>287501294721074</REVISION>
               <MAXHISTORICALORDER>18857</MAXHISTORICALORDER>
               <STALE>false</STALE>
            </IdentifierSet>
            <IdentifierSet>
               <Name>UT-Site-dfe4ddcf-73e5-4c47-b523-50fc47cf9c54</Name>
               <TYPE>SITE</TYPE><DBUID>56181</DBUID>
               <GUID>{EB97CA34-6540-4E86-968F-861A48C72126}</GUID>
               <REVISION>287501294721074</REVISION>
               <MAXHISTORICALORDER>18861</MAXHISTORICALORDER>
               <STALE>false</STALE>
            </IdentifierSet>
         </IdentifierSetList>
      </PutProvisioningDataAndGetIdentifierSetListResponse>
   </s:Body>
```

PutUserImageProvisioningData

Use	Assigns a user image to an existing user. The image appears in the InForm user interface.
Inputs	• Study name. • MedML that contains a single USERREF. • UserImageType. Values: GIF or JPG. • Binary data (via MTOM) for the associated image data. Only one user image can be added per call to this operation.
Outputs	None.
Side effects	Successful result is that the user's image is assigned to the user.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/PutUserImageProvisioningData

Example: PutUserImageProvisioningData

Example request—In this example, the SOAP message is truncated in some areas for brevity.

The corresponding HREF tag value in the Body, shown in bold font, matches the content-ID for the binary data area.

```
POST /pfst45/sdk/provisioning/UserProvisioningService.svc/UserImage HTTP/1.1
MIME-Version: 1.0
Content-Type: multipart/related;
type="application/xop+xml";start="<http://tempuri.org/0>";
boundary="uuid:3af29e87-12a5-44b7-a29e-e7c8e3855efa+id=2";start-
info="application/soap+xml"
Host: localhost Content-Length: 10647 Expect: 100-continue

--uuid:3af29e87-12a5-44b7-a29e-e7c8e3855efa+id=2
Content-ID: <http://tempuri.org/0>
Content-Transfer-Encoding: 8bit
Content-Type: application/xop+xml;charset=utf-8;type="application/soap+xml"

<s:Envelope><s:Header><a:Action s:mustUnderstand="1">
   http://www.phaseforward.com/InForm/2009/01/UserProvisioning/PutUserImageProvi
   sioningData
   </a:Action><a:MessageID>[snip]</a:MessageID><a:ReplyTo>
   <a:Address>[anonymous/snipped]</a:Address></a:ReplyTo><a:To
   s:mustUnderstand="1">
   https://localhost./pfst45/sdk/provisioning/UserProvisioningService.svc/UserIm
   age</a:To>
   <o:Security>[snipped ws-s details from sample]</o:Security></s:Header>
   <s:Body><PutUserImageProvisioningData><TrialName>pfst45</TrialName><MedML>
       <USERREF USERNAME="UserThatGetsImageAssigned"/></MedML>
      <UserImage><xop:Include href="http://tempuri.org/1/634001109046674476"
         xmlns:xop="http://www.w3.org/2004/08/xop/include"/></UserImage>
      <ImageType>GIF</ImageType><Extensions/></PutUserImageProvisioningData></s:
      Body></s:Envelope>

--uuid:3af29e87-12a5-44b7-a29e-e7c8e3855efa+id=2
Content-ID: <http://tempuri.org/1/634001109046674476>
Content-Transfer-Encoding: binary
Content-Type: application/octet-stream
GIF87 [binary image data omitted from sample]
--uuid:3af29e87-12a5-44b7-a29e-e7c8e3855efa+id=2--
```

RemoveAllUsersFromGroups

Use	Removes all users from the specified groups. **Note:** In this release, only removal from SiteGroups is supported.
Inputs	• Study name. • One or more SiteGroups.
Outputs	• IdentifierSet for the SiteGroups referenced. • IdentifierSets for all users removed from any of the site groups.
Side effects	All users are removed from the specified groups. If the InForm system user is part of the group, it will not be removed. A message is logged to the server's event log, stating that the system user was skipped. To remove the system user from a site, use the InForm user interface.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/RemoveAllUsersFromGroups

Example: RemoveAllUsersFromGroups

Example request—This example removes all users from two site groups.

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <RemoveAllUsersFromGroups>
      <TrialName>pfst45</TrialName>
         <MedML>
            <SITEGROUP SITENAME="TestSite-f0b2289a-1db3-4214-8ddb-
            c9230656e52f"></SITEGROUP>
            <SITEGROUP SITENAME="TestSite-a868f070-e39a-4ed4-ad21-
            eb18692e578b"></SITEGROUP>
         </MedML>
      <Extensions/>
   </RemoveAllUsersFromGroups>
</s:Body>
```

Example response—The response lists an identifier set for each user that is removed.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
  <RemoveAllUsersFromGroupsResponse>
     <IdentifierSetList>
        <HasStaleIdentifierSets>false</HasStaleIdentifierSets>
        <IdentifierSet>
           <Name>User-2c9c96be</Name>
           <TYPE>USER</TYPE>
           <DBUID>57604</DBUID>
           <GUID>{285EA4C8-095C-426A-A3D5-CA9C2B764593}</GUID>
           <REVISION>287501791934074</REVISION>
           <MAXHISTORICALORDER>19585</MAXHISTORICALORDER>
           <STALE>false</STALE>
        </IdentifierSet>
        <IdentifierSet>
           <Name>User-5423e8b0</Name>
           <TYPE>USER</TYPE>
           <DBUID>57543</DBUID>
           <GUID>{5575D433-B609-4BE4-9021-8C2D7F42CEE5}</GUID>
           <REVISION>287501791934074</REVISION>
           <MAXHISTORICALORDER>19586</MAXHISTORICALORDER>
           <STALE>false</STALE>
        </IdentifierSet>
        <IdentifierSet>
           <Name>TestSite-a868f070-e39a-4ed4-ad21-eb18692e578b</Name>
           <TYPE>SITE</TYPE>
           <DBUID>57702</DBUID>
           <GUID>{EB93850A-E9F7-4DE3-8663-947445609AD5}</GUID>
           <REVISION>287501791934074</REVISION>
           <MAXHISTORICALORDER>19577</MAXHISTORICALORDER>
           <STALE>false</STALE>
        </IdentifierSet>
        <IdentifierSet>
           <Name>TestSite-f0b2289a-1db3-4214-8ddb-c9230656e52f</Name>
           <TYPE>SITE</TYPE>
           <DBUID>57664</DBUID>
           <GUID>{8913769C-DE8C-42A5-AD18-70F30AEFDBE8}</GUID>
           <REVISION>287501791934074</REVISION>
           <MAXHISTORICALORDER>19573</MAXHISTORICALORDER>
           <STALE>false</STALE>
        </IdentifierSet>
     </IdentifierSetList>
   </RemoveAllUsersFromGroupsResponse>
</s:Body>
```

RemoveUsersFromGroups

Use	Removes users from one or more groups.
Inputs	• Study name. • One or more of: RightsGroup, SiteGroup, QueryGroup, SignatureGroup, ReportingGroup.
Outputs	• IdentifierSet for the groups referenced. • IdentifierSets for all users removed from any of the groups.
Side effects	Specified users are removed from specified groups based on input parameters. If the InForm system user is part of the group, it will not be removed. A message is logged to the server's event log, stating that the system user was skipped. To remove the system user from a site, use the InForm user interface.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/RemoveUsersFromGroups

Example: RemoveUsersFromGroups

Example Request—This example specifies three groups, removing users from each group. .

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body>
   <RemoveUsersFromGroups>
      <TrialName>pfst45</TrialName>
      <MedML>
         <QUERYGROUP GROUPNAME="QGroup-70ddf31c" GROUPDESCRIPTION="QGroup-
         70ddf31c Description>
            <USERREF USERNAME="User-59708ea0" /></QUERYGROUP>
         <REPORTINGGROUP GROUPNAME="RepGroup-ee2d0825"
         GROUPDESCRIPTION="RepGroup-ee2d0825 Description">
            <USERREF USERNAME="User-49ada9ee" />
            <USERREF USERNAME="User-59708ea0" />
         </REPORTINGGROUP>
         <REPORTINGGROUP GROUPNAME="RepGroup-1cb432a9"
         GROUPDESCRIPTION="RepGroup-1cb432a9 Description">
            <USERREF USERNAME="User-49ada9ee" />
            <USERREF USERNAME="User-59708ea0" />
         </REPORTINGGROUP>
      </MedML><Extensions/>
   </RemoveUsersFromGroups>
</s:Body>
```

Example Response—The response contains an identifier set list that includes an identifier set for each user and for each group.

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <RemoveUsersFromGroupsResponse>
      <IdentifierSetList>
         <HasStaleIdentifierSets>false</HasStaleIdentifierSets>
         <IdentifierSet>
            <Name>User-49ada9ee</Name>
            <TYPE>USER</TYPE>
            <DBUID>59290</DBUID>
            <GUID>{FE536265-71C3-4B48-A1D3-B564F0A22736}</GUID>
            <REVISION>287501984206074</REVISION>
            <MAXHISTORICALORDER>20352</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>User-59708ea0</Name>
            <TYPE>USER</TYPE>
            <DBUID>59351</DBUID>
            <GUID>{F92E749A-7E56-4A7B-91EF-0CF9F1A3502D}</GUID>
            <REVISION>287501984206074</REVISION>
            <MAXHISTORICALORDER>20353</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>QGroup-70ddf31c</Name>
            <TYPE>QUERYGROUP</TYPE>
            <DBUID>59411</DBUID>
            <GUID>{CAFBF8FB-053B-447E-9326-7FD1C6D17F29}</GUID>
            <REVISION>287501984206074</REVISION>
            <MAXHISTORICALORDER>20336</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>RepGroup-1cb432a9</Name>
            <TYPE>REPORTINGGROUP</TYPE>
            <DBUID>59429</DBUID>
            <GUID>{7264C33B-523C-4BD0-B602-C650A275D3D1}</GUID>
            <REVISION>287501984206074</REVISION>
            <MAXHISTORICALORDER>20346</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
         <IdentifierSet>
            <Name>RepGroup-ee2d0825</Name>
            <TYPE>REPORTINGGROUP</TYPE>
            <DBUID>59420</DBUID>
            <GUID>{6D116B3E-F84C-459D-8EC1-BDD63E452F76}</GUID>
            <REVISION>287501984206074</REVISION>
            <MAXHISTORICALORDER>20341</MAXHISTORICALORDER>
            <STALE>false</STALE>
         </IdentifierSet>
      </IdentifierSetList>
   </RemoveUsersFromGroupsResponse>
</s:Body>
```

UpdateSiteNameAndMnemonic

Use	Changes the name and mnemonic for a single site.
Inputs	• Study name. • Site mnemonic of existing site. • New site mnemonic. • New site name.
Outputs	None.
Side effects	For the specified site, the old/existing mnemonic will be replaced by the new mnemonic, and the old/existing site name will be replaced by the new site name.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/UpdateSiteNameAndMnemonic

Example: UpdateSiteNameAndMnemonic

Example request

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <UpdateSiteNameAndMnemonic>
      <TrialName>pfst45</TrialName>
      <OldSiteMnemonic>(01)</OldSiteMnemonic>
      <NewSiteMnemonic>(01-a)</NewSiteMnemonic>
      <NewSiteName>(01-a) Hospital (Division A)</NewSiteName><Extensions/>
   </UpdateSiteNameAndMnemonic>
</s:Body>
```

VerifyPassword

Use	Verifies the validity of a user name and password combination.
Inputs	• Study name. • User name. • Password.
Outputs	• True—User name and password are valid and the account is active. • False—User name and password are invalid and/or the account is not active.
Side effects	Invalid password combinations trigger standard InForm password processing logic and can trigger deactivation of the account.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/VerifyPassword

Example: VerifyPassword

Example request

The example shows the Body of the request. It must be preceded by the SOAP header that contains the SOAP action listed in the table for this operation.

```
<s:Body >
   <VerifyPassword>
      <TrialName>PFST45</TrialName>
      <Credentials>
         <UserName>aroberts</UserName>
         <Password>changeme</Password>
      </Credentials>
      <Extensions/>
   </VerifyPassword>
</s:Body>
```

Example response

The body of the response is preceded by the SOAP header, which is not shown in this example.

```
<s:Body>
   <VerifyPasswordResponse>
      <VerifyPasswordResult>true</VerifyPasswordResult>
   </VerifyPasswordResponse>
</s:Body>
```

CHAPTER 6

Use Cases

In this chapter

Users

Creating users

To create new users:

- Use the **PutProvisioningData** operation.

To create new users and return an identifier set for each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.**

Usage—for both operations:

- Specify one or more MedML <USER> elements for non-existent users.
- If the USERNAME you specify does not already exist, the user is automatically created.
- If you specify a USERNAME that already exists, the existing user is updated with the values you specify for the other fields.
- USERNAME values are case-sensitive.

Users and Rights groups

- If you are using the InForm 5.5 SP0b or later software release, new users that you add with the User Management Interface must have an associated rights group to use the InForm application. For more information, see ***Creating a rights group*** (on page 74).

Values: USER element—Create user		
Field	Required?	Comments
USERNAME	Yes	
USERTYPE	Yes	
FIRSTNAME	No	
LASTNAME	No	
DISPLAYNAME	No	
DESCRIPTION	No	
TITLE	No	
ADDRESS	No	
ADDRESS2	No	
CITY	No	
STATE	No	STATE and PROVINCE are mutually exclusive.
PROVINCE	No	STATE and PROVINCE are mutually exclusive.

Values: USER element—Create user		
Field	**Required?**	**Comments**
ZIPCODE	No	ZIPCODE and POSTCODE are mutually exclusive.
POSTCODE	No	ZIPCODE and POSTCODE are mutually exclusive.
COUNTRY	No	
PHONE	No	
ALTPHONE	No	
FAX	No	
EMAIL	No	
BEEPER	No	
HOMESCREENURL	No	
ACTIVESTATE	No	Active/Inactive, defaults to Inactive.
DELETESTATE	No	Terminated/Reinstate.
PASSWORD	No	
USERMUSTRESETPASSWORD	No	
USERDATEFORMAT	No	
PRODUCTLOCALE	Yes	Required when creating new USER elements. Values: ja-JP or en-US. (This is an InForm 5.0 and above restriction.) When the User Management Interface is used with an InForm 4.6 installation, this attribute is ignored.
STUDYLOCALE	Yes	Required when creating new USER elements. Value must be a study locale that is defined in some study version in the trial. When the User Management Interface is used with an InForm 4.6 installation, this attribute is ignored.

Updating users

To update existing users:

- Use the **PutProvisioningData** operation.

To update existing users and return identifier sets for the updated users:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note:** **Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.**

Usage—for both operations:

- Specify one or more MedML <USER> elements for existing users.
- If the USERNAME you specify exists, the user is updated.
- USERNAME values are case-sensitive. If the name you specify does not exactly match an existing name, and you also specify USERTYPE, a new user is created.
- Updates to both user information (for example, a title) and user state (for example, ActiveState or PasswordReset) can be performed simultaneously.

Values: USER element—Update user		
Field	Required?	Comments
USERNAME	Yes	
USERTYPE	No	
FIRSTNAME	No	
LASTNAME	No	
DISPLAYNAME	No	
DESCRIPTION	No	
TITLE	No	
ADDRESS	No	
ADDRESS2	No	
CITY	No	
STATE	No	STATE and PROVINCE are mutually exclusive.
PROVINCE	No	STATE and PROVINCE are mutually exclusive.
ZIPCODE	No	ZIPCODE and POSTCODE are mutually exclusive.

Values: USER element—Update user		
Field	Required?	Comments
POSTCODE	No	ZIPCODE and POSTCODE are mutually exclusive.
COUNTRY	No	
PHONE	No	
ALTPHONE	No	
FAX	No	
EMAIL	No	
BEEPER	No	
HOMESCREENURL	No	
ACTIVESTATE	No	Active/Inactive, defaults to Inactive.
DELETESTATE	No	Terminated/Reinstate.
PASSWORD	No	If you specify a new password for an existing user, the password is changed and the user account remains unchanged.
USERMUSTRESETPASSWORD	No	
USERDATEFORMAT	No	
PRODUCTLOCALE	No	If the client does not specify a value, the existing value is used. When the User Management Interface is used with an InForm 4.6 installation, this attribute is ignored.
STUDYLOCALE	No	If the client does not specify a value, the existing value is used. When the User Management Interface is used with an InForm 4.6 installation, this attribute is ignored.

Changing passwords

To change passwords for existing users:

- Use the **PutProvisioningData** operation.

To change passwords for existing users and return an identifier set for each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

Note: **Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.**

Usage—for both operations:

- Specify one or more MedML <USER> elements with existing USERNAME values and updated PASSWORD field values.
- Passwords specified with the API are subject to password policies defined in the InForm configuration (for example, at least one numerical character is required in the password). The USERNAME value is case-sensitive and must match exactly as described in ***Updating users*** (on page 62).

Resetting passwords

To enable or modify the password reset option for existing users:

- Use the **PutProvisioningData** operation.

To enable or modify the password reset option for existing users, and return an identifier set for each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note:** Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <USER> elements using existing USERNAME values and desired USERMUSTRESETPASSWORD values.
- The USERNAME value is case-sensitive and must match exactly as described in ***Updating users*** (on page 62).

Verifying passwords

To validate InForm user credentials for a single user:

- Use the **VerifyPassword** (on page 57) operation to facilitate the external validation of credentials.

Password verification is subject to the same business logic as InForm authentication and results in account deactivation upon multiple failed attempts based on the configured InForm password policy.

Use	Verifies the validity of a user name and password combination.
Inputs	• Study name. • User name. • Password.
Outputs	• True—User name and password are valid and the account is active. • False—User name and password are invalid and/or the account is not active.

Side effects	Invalid password combinations trigger standard InForm password processing logic and can trigger deactivation of the account.
SOAP action	http://www.phaseforward.com/InForm/2009/01/UserProvisioning/ VerifyPassword

Changing the state of a user

To change the account state for users:

- Use the **PutProvisioningData** operation.

To change the account state for users and return an identifier set for each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.**

Usage—for both operations:

- Specify one or more MedML <USER> elements with existing USERNAME values and updated ACTIVESTATE and DELETESTATE values.
- The user account state follows the same business logic as the InForm user interface:
  - An ACTIVESTATE value of "true" sets a user status to Active, and a value of "false" sets a user status to Inactive.
  - A DELETESTATE value of "true" sets the user to Terminated.
- The USERNAME value is case-sensitive and must match exactly as described in ***Updating users*** (on page 62).

Retrieving user names

To retrieve a list of all user names in the study:

- Use the ***GetUserNames*** (on page 44) operation to facilitate retrieval of users that have been added to the study.

  You can use a filter to retrieve a subset of users in the study. A percent sign (%) indicates a wildcard. For example, "S%" returns any user name that starts with S or s. Differences in case are ignored. If the study contains user names that differ only in letter case, all matching names are retrieved.

Retrieving user identifier information

To retrieve a fresh IdentifierSetList of type USER, without creating or updating users:

- Use the ***GetIdentifierSetList*** (on page 36) operation and specify one or more IdentifierSet elements with TYPE=User, in addition to the user name. Leave the other properties empty or to their default values.

Values: IdentifierSet		
Field	Required?	Comments
Name	Yes	
TYPE	Yes	

Retrieving user attribute information

To retrieve a list of users that contains details about each user:

- Use the ***GetUserDetails*** (on page 42) operation to retrieve user attributes as MedML data.

Insert an image for a user

To add a user image that will appear in the InForm user interface:

- Use the **PutUserImageProvisioningData** operation by specifying the MedML that contains a single USERREF.

Values: PutUserImageProvisioningData		
Field	Required?	Comments
TrialName	Yes	
USERREF USERNAME	Yes	Name of user, as MedML. **Note:** Only one user image can be added per call.
<UserImage>	Yes	HREF that points to the image.
<ImageType>	Yes	Values: GIF, JPG.
<ImageData>	Yes	Binary data (via MOTM) for the associated image data.

Sites

Creating sites

To create new sites:

- Use the **PutProvisioningData** operation.

To create new sites and return an identifier set for each site:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <SITE> elements for non-existent sites.
- If the site NAME you specify does not already exist, the site is automatically created.

Values: SITE element—Create site		
Field	Required?	Comments
NAME	Yes	
MNEMONIC	Yes	
ADDRESS	No	
ADDRESS2	No	
CITY	No	
STATE	No	STATE and PROVINCE are mutually exclusive.
PROVINCE	No	STATE and PROVINCE are mutually exclusive.
ZIPCODE	No	ZIPCODE and POSTCODE are mutually exclusive.
POSTCODE	No	ZIPCODE and POSTCODE are mutually exclusive.
COUNTRY	No	
PHONE	No	
ALTPHONE	No	
FAX	No	
EMAIL	No	
BEEPER	No	

Values: SITE element—Create site

Field	Required?	Comments
TIMEZONE	Yes	For the format of this field, see the InForm *Utilities Guide*.
STARTDATE	Yes	For the format of this field, see the InForm *Utilities Guide*.
ENDDATE	No	For the format of this field, see the InForm *Utilities Guide*.
SITEDATEFORMAT	No	
STUDYLOCALE	Yes	Clients must include this attribute when creating new SITE elements. If present, this must be a study locale that is defined in some study version of the trial. When the User Management Interface is used with an InForm 4.6 installation, this attribute is ignored. **Note:** A site's STUDYLOCALE cannot be changed after any data is entered for the site. This means that the value can be changed until the first Submit of a screening form for the first candidate subject.
USERNAMEORDER	Yes	Clients must include this attribute when creating new SITE elements. Specifies the order of a user's first and last names in the signature affidavit. Possible values: • "F,L" first name followed by last name. • "L,F" last name followed by first name. **Note:** No space follows the comma. When the User Management Interface is used with an InForm 4.6 installation, this attribute is ignored.

Updating sites

To update existing sites:

- Use the **PutProvisioningData** operation.

To update existing users and return an identifier set for each site:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note:** Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <SITE> elements for existing sites.
- If the site NAME you specify exists, the site is automatically updated.
- Site NAME values are case-sensitive. If the name you specify does not exactly match an existing name, and you also specify MNEMONIC, STARTDATE, and TIMEZONE, a new site is created.

> **Note:** To change the Mnemonic, use ***UpdateSiteNameAndMnemonic*** (on page 56).

Values: SITE element—Update site		
Field	Required?	Comments
NAME	Yes	
ADDRESS	No	
ADDRESS2	No	
CITY	No	
STATE	No	STATE and PROVINCE are mutually exclusive.
PROVINCE	No	STATE and PROVINCE are mutually exclusive.
ZIPCODE	No	ZIPCODE and POSTCODE are mutually exclusive.
POSTCODE	No	ZIPCODE and POSTCODE are mutually exclusive.
COUNTRY	No	
PHONE	No	
TIMEZONE	No	
STARTDATE	No	
ENDDATE	No	
SITEDATEFORMAT	No	

Values: SITE element—Update site		
Field	**Required?**	**Comments**
STUDYLOCALE	No	A site's STUDYLOCALE cannot be changed after any data is entered for the site. This means that the value can be changed until the first Submit of a screening form for the first candidate subject. When updating a site, include this attribute only if no data has been entered for that site's study.
USERNAMEORDER	No	If the client does not specify a value, the existing value is used. When the User Management Interface is used with an InForm 4.6 installation, this attribute is ignored.

Changing the site name and mnemonic of an existing site

To change the site name and/or the mnemonic for an existing site:

- Use the **UpdateSiteNameAndMnemonic** (on page 56) operation, specifying the study name and, optionally, the existing mnemonic for the site.

Values: Update site name and mnemonic		
Field	**Required?**	**Comments**
TrialName	Yes	
OldSiteMnemonic	Yes	Must match a site's mnemonic in the study.
NewSiteMnemonic	Yes	Must be unique in the study.
NewSiteName	Yes	Must be unique in the study.

Adding users to a site

To add users to one or more sites:

- Use the **PutProvisioningData** operation.

To add users to one or more sites and return an identifier set for each user and site:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <SITEGROUP> elements that contain the site name and list of users to be added to the site.
- The SITENAME and USERNAME values are case-sensitive and must match exactly or an error is returned.

Values: SITEGROUP element—Add user to a site		
Field	Required?	Comments
SITENAME	Yes	
USERREF	No	One or more user names to add to the site per the MedML schema.

Getting a list of users and adding them to a site

To get a list of users of a particular type, and add those users to a site:

1. Use the **GetUserNamesByType** (on page 45) operation to return a list of users of the type you specify.
2. Use the **AddUsersToSite** (on page 34) operation to add the users to a specified site.

Removing users from a site

To remove users from one or more sites:

- Use the **RemoveUsersFromGroups** operation to specify one or more MedML <SITEGROUP> elements that contain the site name and list of users to be removed from the site.

  The SITENAME and USERNAME values are case-sensitive and must match exactly or an error is returned.

Values: SITEGROUP element—Remove user from a site		
Field	Required?	Comments
SITENAME	Yes	
USERREF (child element)	Yes	One or more user names to remove from the site per the MedML schema.

Removing all users from a site

To remove all users from one or more sites:

- Use the **RemoveAllUsersFromGroups** (on page 52) operation to specifying one or more MedML <SITEGROUP> elements that contain the name of the site from which all users should be removed.

  The SITENAME value is case-sensitive and must match exactly or an error is returned.

Values: SITEGROUP element—Remove all users from a site		
Field	Required?	Comments
SITENAME	Yes	Must match a site name in the study.

Changing the site study version

To update site study version:

- Use the **PutProvisioningData** operation.

To update the site study version and return identifier sets for the sites affected:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more <STUDYVERSIONSITE> elements that contain the name of the site and the new study version.
- The SITEMNEMONIC and VERSIONDESCRIPTION values are case-sensitive and must match exactly or an error is returned.

Values: STUDYVERSIONSITE element—Change study version for a site		
Field	Required?	Comments
VERSIONDESCRIPTION	Yes	Value is obtained from call to GetStudyVersions.
SITEMNEMONIC	Yes	
ACCEPTDATE	Yes	For the format of this field, see the InForm *Utilities Guide*.
REASON	No	

Retrieving site information

To retrieve a list of sites in a study, without creating or updating the sites:

- Use the **GetIdentifierSetList** (on page 36) operation to retrieve an IdentifierSet of type SITE or SITE elements that exist in the study.

Values: IdentifierSet		
Field	Required?	Comments
Name	Yes	
TYPE	Yes	

Rights groups

Creating a rights group

To create a rights group:

- Use the **PutProvisioningData** operation.

To create a rights group and return an identifier set for each group and user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note:** Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <RIGHTSGROUP> elements for non-existent rights groups.
- If the GROUPNAME you specify does not already exist, the rights group is automatically created.

Values: RIGHTSGROUP—Create rights group		
Field	Required?	Comments
GROUPNAME	Yes	
GROUPDESCRIPTION	No	
RIGHTREF	No	One or more right refnames associated with the rights group.
USERREF	No	One or more user names to add to the rights group. A user can belong to only one rights group. If you specify a user who already belongs to a different rights group, the user will be removed from that group and added to the rights group specified by this operation.
ITEMGROUPREF	No	One or more item group ref names associated with the rights group.

Updating a rights group

To update existing rights groups:

- Use the **PutProvisioningData** operation.

To update existing rights groups and return an identifier set for each rights group and user

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note:** Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <RIGHTSGROUP> elements for existing right groups.
- If the GROUPNAME you specify exists, the rights group is automatically updated.
- GROUPNAME values are case-sensitive. If the name you specify does not exactly match an existing name, a new rights group is created.
- Values for the fields are identical to those used when creating a rights group. For more information, see ***Creating a rights group*** (on page 74).

Changing a users rights group

You add users to new or existing rights groups during the creation or updating of the rights group. For more information, see ***Creating a rights group*** (on page 74) and ***Updating a rights group*** (on page 75).

InForm users are only associated with a single rights group, which you can change. For more information, see ***Updating a rights group*** (on page 75).

Removing a user from a rights group

To remove a user from a rights group:

- Use the **RemoveUsersFromGroups** operation to specify one or more MedML <RIGHTSGROUP> elements containing the rights group name and list of users to be removed from the group.

  Specifying users:

  - If the list of users includes a user that exists in the study, but not in the group you specify, the request to remove that particular user from the group is ignored. The users that are group members are removed from the group.
  - If you include a user that is not in the study, a SOAP fault occurs with Invalid Data as the Fault Code, and an error message states that the user name was not found in the study. For more information, see ***How requests are processed*** (on page 24).

GROUPNAME values are case-sensitive. If the GROUPNAME you specify does not exist, an error is returned.

Values: RIGHTSGROUP—Remove user from rights group		
Field	Required?	Comments
GROUPNAME	Yes	
USERREF (child element)	Yes	One or more usernames to remove from the group.

Retrieving group identifier information

To retrieve a list of identifier set information for groups in the study without creating or updating groups:

- Use the **GetIdentiferSetList** (on page 36) operation to retrieve an IdentifierSet of type RIGHTSGROUP, QUERYGROUP, SIGNATUREGROUP, or REPORTINGGROUP.

Values: IdentifierSet		
Field	Required?	Comments
Name	Yes	
TYPE	Yes	

Query groups

Creating a query group

To create a query group:

- Use the **PutProvisioningData** operation.

To create a query group and return an identifier set for each group and user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <QUERYGROUP> elements for non-existent query groups.
- If the GROUPNAME you specify does not exist, the query group is automatically created.
- GROUPNAME values are case-sensitive. If you specify a GROUPNAME that already exists, the existing group is updated.

Values: QUERYGROUP element—Create query group		
Field	Required?	Comments
GROUPNAME	Yes	
GROUPDESCRIPTION	No	
USERREF (child element)	No	One or more user names to add to the query group. A user can belong to only one query group. If you specify a user who already belongs to a different query group, the user will be removed from that group and added to the query group specified by this operation.

Updating a query group

To update a query group:

- Use the **PutProvisioningData** operation.

To create a query group and return an identifier set for the group and each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <QUERYGROUP> elements for existing query groups.
- If the GROUPNAME you specify exists, the query group is automatically updated.
- GROUPNAME values are case-sensitive. If the name you specify does not exactly match an existing name, a new query group is created.

Values: QUERYGROUP element—Update query group		
Field	Required?	Comments
GROUPNAME	Yes	
GROUPDESCRIPTION	No	
USERREF (child element)	Yes	One or more user names to add to the query group. A user can belong to only one query group. If you specify a user who already belongs to a different query group, the user will be removed from that group and added to the query group specified by this operation.

Changing a users query group

You add users to new or existing query groups during the creation or updating of the query group. For more information, see ***Creating a query group*** (on page 77) and ***Updating a query group*** (on page 77).

InForm users are associated only with a single query group, which you can change. For more information, see ***Updating a query group*** (on page 77).

Removing a user from a query group

To remove a user from one or more query groups:

- Use the **RemoveUsersFromGroups** operation to specify one or more MedML <QUERYGROUP> elements containing the query group name and list of users to be removed from the group

  GROUPNAME values are case-sensitive. If the name you specify does not exist, an error is returned.

Values: QUERYGROUP element—Remove user from query group		
Field	Required?	Comments
GROUPNAME	Yes	If the GROUPNAME you specify does not exist, the operation returns an error.
USERREF (child element)	Yes	One or more user names to remove from the query group.

Signature groups

Associating a user with a signature group

To associate a user with a signature group:

- Use the **PutProvisioningData** operation.

To associate a user with a signature group and return an identifier set for the group and each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note: Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.**

Usage—for both operations:

- Specify one or more <SIGNATUREGROUP> elements containing the signature group name and list of users to be added to the group. If the specified signature group does not exist, it will be created.
- InForm users are associated only with a single signature group, which is updated based on the GROUPNAME that you specify.
- GROUPNAME values are case-sensitive. If the name you specify does not exist, an error is returned.

Values: SIGNATUREGROUP element—Associate a user with a signature group

Field	Required?	Comments
GROUPNAME	Yes	
USERREF (child element)	Yes	One or more user names to add to the group. A user can belong to only one signature group. If you specify a user who already belongs to a different signature group, the user will be removed from that group and added to the signature group specified by this operation.

Removing a user from a signature group

To remove users from a signature group:

- Use the RemoveUsersFromGroups operation to specify one or more MedML.
- Specify one or more <SIGNATUREGROUP> elements containing the signature group name and list of users to be removed from the group.
- GROUPNAME values are case-sensitive. If the GROUPNAME you specify does not exist, an error is returned.

Values: SIGNATUREGROUP element—Remove user from signature group		
Field	Required?	Comments
GROUPNAME	Yes	
USERREF (child element)	Yes	One or more user names to remove from the group.

Reporting groups

Creating a reporting group

To create a reporting group:

- Use the **PutProvisioningData** operation.

To create a reporting group and return an identifier set for the group and each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note:** Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <REPORTINGGROUP> elements for non-existent reporting groups.
- If the GROUPNAME you specify does not exist, the reporting group is automatically created.

Values: REPORTINGGROUP element—Create reporting group		
Field	Required?	Comments
GROUPNAME	Yes	
GROUPDESCRIPTION	No	
USERREF (child element)	No	One or more user names to add to the group.

Updating a reporting group

To update a reporting group:

- Use the **PutProvisioningData** operation.

To update a reporting group and return an identifier set for the group and for each user:

- Use the **PutProvisioningDataAndGetIdentifierSets** (on page 49) operation.

> **Note:** Use PutProvisioningDataAndGetIdentiferSets if the returned identifier sets are used within the client. The operation is significantly slower than PutProvisioningData.

Usage—for both operations:

- Specify one or more MedML <REPORTINGGROUP> elements for existing reporting groups. Values for the fields are identical to those used when creating a reporting group.
- If the GROUPNAME you specify exists in the system, the reporting group is automatically updated.
- GROUPNAME values are case-sensitive. If the name you specify does not exactly match an existing name, a new rights group is created.

Values: REPORTINGGROUP element—Update reporting group		
Field	Required?	Comments
GROUPNAME	Yes	
GROUPDESCRIPTION	No	
USERREF (child element)	Yes	One or more user names to add to the group.

Adding a user to a reporting group

You add users to new or existing reporting groups during the creation or removal of the reporting group. For more information, see ***Creating a reporting group*** (on page 81) and ***Removing a user from a reporting group*** (on page 82).

Removing a user from a reporting group

To remove a user from one or more reporting groups:

- Use the **RemoveUsersFromGroups** operation to specify one or more MedML <REPORTINGGROUP> elements containing the reporting group name and list of users to be removed from the group.

  GROUPNAME values are case-sensitive. If the GROUPNAME you specify does not exist, an error is returned.

Values: REPORTINGGROUP element—Remove user from reporting group		
Field	Required?	Comments
GROUPNAME	Yes	
USERREF (child element)	Yes	One or more user names to remove from the group.

Study versions

Retrieving study versions

To retrieve the list of all study versions defined in the study:

- Use the **GetStudyVersions** (on page 41) operation.

  Study versions are returned in the order they were created with the latest study version appearing last.

To retrieve only the latest study version:

- Use the **GetLatestStudyVersion** (on page 39) operation.

CHAPTER 7

Error handling

In this chapter

About error handling

Descriptive error messages are returned to the web service client. Errors are written to the InForm server event log and include the following information:

- Date and time that the error occurred.
- Error Level.
- Error Number.
- Error Description.

Error messages are published to the Windows Event log under the name “InFormProvisioning.”

Errors are returned through HTTP 1.1 and SOAP 1.2 compliant faults.

HTTP errors

Requests that exceed the allowable maximum request size will be rejected by the server. The error "HTTP 400 Bad Request" is returned to the client and further details related to the error are written to the Windows event log.

SOAP errors

SOAP fault sub-codes defined by the API are interpreted in the http://www.phaseforward.com/InForm/2009/01/UserProvisioning namespace.

Basic summary information is included when available to provide context regarding the cause of the fault. For more information, see ***Examples: SOAP errors*** (on page 89). Detailed error information is written to the server event log.

Fault subcode	Notes
InvalidTrial	The specified study does not exist on the server.
TrialNotStarted	The specified study is not started on the server.
ServerNotStarted	The InForm server has not been started.
InvalidData	Reference to invalid entity (for example, bad site or user name in sitegroup), invalid characters in the input, or schema validation failure.
ConfigurationError	API configuration error.
TransactionTimeout	Request processing has exceeded the maximum transaction processing window, which resulted in a failure and rollback of processing.
TrialUrlMismatch	Study name that is specified in the request does not match the name specified in the URL.
FailedAuthentication	Invalid credentials or non-active account.
InternalError	Internal, unexpected error occurred.

Examples: SOAP errors

Fault subcode	Reason text
InvalidData	Required attribute USERNAME not found.
InvalidData	USERNAME for "username" does not exist.
InvalidData	SITE "sitename" does not exist.
InvalidData	A user name must contain only alphabetic or numeric characters or underscores and may not contain spaces.
InvalidData	The attribute "myattribute" on the element '{PhaseForward-MedML-Inform4}USER' is not defined in the DTD/Schema.
InvalidData	PutProvisioningData failed. There are {0} USER elements without USERTYPE specified. USERTYPE is required when creating a new USER.
InvalidData	PutProvisioningData failed. There are {0} SITE elements without MNEMONIC specified. MNEMONIC is required when creating a new SITE.
InvalidData	An error occurred attempting to add missing USERTYPE to MedML USER element for USERNAME {0}.
InvalidData	An error occurred attempting to add missing MNEMONIC to MedML SITE element for SITE {0}.
InvalidData	PutProvisioningData MedML schema validation failed as follows: {0} {1}.
InvalidData	RemoveUsersFromGroups MedML element does not contain any group elements.
InvalidData	RemoveAllUsersFromGroups MedML element does not contain any SITEGROUP elements.
InvalidData	RemoveAllUsersFromGroups request does not specify a study name.
InvalidData	GetUserNames request does not specify a study name.
InvalidData	VerifyPassword request does not specify a study name.
InvalidData	GetStudyVersions request does not specify a study name.

Example of SOAP fault

```
<s:Body>
   <s:Fault>
      <s:Code>
         <s:Value>s:Sender</s:Value>
         <s:Subcode>
            <s:Value
            xmlns:a="http://www.phaseforward.com/InForm/2009/01/UserProvisioning"
            >a:InvalidData</s:Value>
         </s:Subcode>
      </s:Code>
      <s:Reason>
         <s:Text xml:lang="en-US">Required attribute USERNAME not found</s:Text>
      </s:Reason>
   </s:Fault>
</s:Body>
```

CHAPTER 8

Troubleshooting

In this chapter

Processing the trial

For the User Management Interface software to perform processing, you must specify the MedML Installer utility server and the site server as required for standard InForm deployment.

To specify the servers, open a Command Prompt window on the server where the InForm software is installed, and issue the following commands:

```
 > pfadmin setserver medmlinstaller <trial_name>
<InForm_appplication_server_name>

 > pfadmin setserver site <trial_name> <InForm_application_server name>
```

Enable WSDL

The WSDL generation is disabled by default. To enable it, you need to make the following changes to the web.config file under the Provisioning directory of your install.

- Uncomment the following line:

  <endpoint contract="IMetadataExchange" binding="mexHttpBinding" address="mex" />

- Find the line <serviceMetadata httpGetEnabled="false" /> and change it to <serviceMetadata httpGetEnabled="true" />.

MedML restrictions

The InForm application allows MedML to be installed even if you do not specify the namespace for elements, which bypasses schema validation. As a result, some MedML files that would have failed schema validation (for example, due to invalid case) can successfully be processed. The API enforces strict schema validation and as a result some existing MedML content might need to be altered so that it works with the User Management Interface software.

Oracle listener refuses connections or unavailable

If a large number of connections (more than the value of PROCESSES) to Oracle are opened and closed within a short period of time (~10 min), the listener service might assume that the Oracle instance is unable to accept new connections and block new connections until it receives an update on load and connection status from the Oracle instance.

Symptoms

One or more of the following errors might occur:

- InForm error in the Application event log:

```
Action: Connect to database
DSN: <pfst45>
Error: SQL Connection Error #1 State: S1000 Message: [Oracle][ODBC][Ora]ORA-
12520: TNS:listener could not find available handler for requested type of
server
SQL Connection Error #2 State: S1000 Message: [Oracle][ODBC][Ora]ORA-12520:
TNS:listener could not find available handler for requested type of server
```

- Error in Oracle Listener Log (default location: <oracle_home>\NETWORK\log\listener.log):

```
01-SEP-2009 10:24:03 *
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=Trial1.world)(CID=(PROGRAM=C:\W
INDOWS\system32\dllhost.exe)(HOST=RDNEPTUNQ005)(USER=PfUSR_RDNEPTUNQ005))) *
(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=2046)) * establish *
Trial1.world * 12520
TNS-12520: TNS:listener could not find available handler for requested type
of server
```

Resolution

Increase the value of PROCESSES for the Oracle instance to allow room for the number of connections that will be opened/closed within a 10-minute period.

For more information about the Oracle listener and connection information, see the Oracle MetaLink article 240710.1 on the Oracle website.