iAccess for Maconomy

Install Guide
2017

Edited by

Anders Hessellund
Peter Enevoldsen

While Deltek has attempted to verify that the information in this document is accurate and complete, some typographical or technical errors may exist. The recipient of this document is solely responsible for all decisions relating to or use of the information provided herein.

The information contained in this publication is effective as of the publication date below and is subject to change without notice.

This edition published May 2017.

Contents

Revision History

Date	Author	Notes
2015-12-28	AH	First draft of iAccess 1.2 documentation. Installation notes not included yet.
2016-01-08	AH	Terminology aligned with other iAccess products, e.g., *View* replaces *Screen*, *Leftnav* replaces *Sidebar*. Incorporated installation notes.
2016-01-20	AH	Aligned with both iAccess version 1.2.0 and 1.2.1.
2016-01-31	PE	Adding SSO preferences.
2016-02-07	AH	Aligned documentation with iAccess 1.3.
2016-02-19	PE	Extended the upgrade path with relation to removal of default preferences.
2016-08-17	AH	Merged “Install Guide” and “Extension Manual” into a single document.
2016-08-30	AH	Updated the merged document with review comments, suggestions and corrections.
2016-11-09	MC	Corrections
2016-11-10	AH	API changed from 2.0.0 to 3.0.0. Compatibility narrowed to 2.3GA. Updated Migration Guide and Troubleshooting Section.
2016-11-10	CC	Included additional SPNEGO SSO browser setup instructions.
2017-03-10	AH	Aligned with iAccess 2.0

Chapter 1

Introduction

The following document serves as an introduction to the iAccess for Maconomy product. The target audience is technical consultants and partners that need to install, extend, and maintain iAccess. In the first part, we will describe core concepts of the iAccess architecture. The second part describes how it can be installed using MConfig. The third part describes how it can be extended using the Maconomy Extender. In the fourth part, we will provide an overview of the current extension points. This part can be used as reference when working with iAccess extensions. Finally, the fifth part contains a migration guide, and a troubleshooting guide. Both of these guides should be particularly useful when upgrading an existing iAccess installation.

We also refer the reader to our Kona space, *iAccess for Maconomy*, where Product Management, and the Development Team will answer questions, and discuss feature requests for the product.

Chapter 2

iAccess Architecture

Briefly described, iAccess for Maconomy is an HTML5 web client. It is a lightweight user interface supplement to the existing Workspace Client. The backend is Maconomy, specifically the new RESTful web services exposed from Maconomy version 2.2 [2]. In this section, we will give a cursory overview of the technical architecture.

2.1 Technical Architecture

Figure 13 shows the high-level architecture of a Maconomy system with iAccess. This setup resembles the traditional Maconomy architecture with a few exceptions. In the following section, we will describe the core components involved and the purpose of each of those.

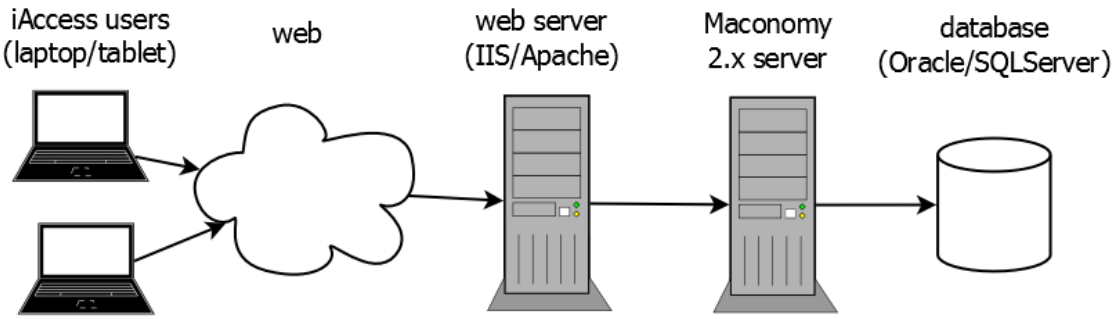

Figure 13: Architectural overview.

Maconomy 2.x server and database : iAccess for Maconomy is available from Maconomy 2.2. It does not impose any specific requirements on the database, but it does require a Maconomy 2.2 server and corresponding RESTful API [2]. See the following section for more details on the required web services.

Web server (IIS/Apache) : One or more web servers are required to serve both the static and dynamic content of iAccess. Static content such as HTML, JavaScript, CSS files, and so on are placed directly on the web server. Dynamic content such as specifications, files, and data are retrieved from the Maconomy server, but the web server in this case acts as proxy. Using the web server as proxy prevents cross-origin (CORS) issues on the client side. The web server is also required for encryption and compression of client-web server communication.

iAccess Clients : The iAccess clients can be located both on the internal network or on the open Internet depending on the web server configuration and exposure. Furthermore, clients can run iAccess on different devices such as laptops with the main browsers (IE, Chrome, and Safari), as well as on iOS and Android tablets.

2.1.1 RESTful Web Service API

As mentioned previously, iAccess for Maconomy uses the RESTful Web Service API which was introduced in Maconomy 2.2. This is *not* the same thing as the existing MScript web services. Please see the RESTful Web Services documentation for more information [2]. For now, we will just briefly list the three web service endpoints that iAccess uses and what they are used for:

/containers : The *containers* endpoint delivers both metadata and data for the containers exposed by the Maconomy 2x server. Metadata include specifications of the names, actions, fields, and foreign keys exposed by different containers. Data include the actual filter-, card-, and table-data stored in the underlying database as well as information on which actions are enabled.

/filedrop : The *filedrop* endpoint is used to upload files such as receipt attachments on expense sheets.

/configurations : The *configurations* endpoint was introduced in Maconomy 2.2.2 and is used by iAccess 1.2 and onwards. This endpoint is used to retrieve JSON specifications from the Maconomy server, specifically the application specification (`application.json`) which configures iAccess. These specifications are the foundation of the iAccess extensibility model.

/auth : The *auth* endpoint was introduced in Maconomy 2.3GA and is used by iAccess 2 and onwards. This endpoint is used to obtain login tokens for 3rd party integrations such as Business Objects.

/environment : The *environment* endpoint was introduced in Maconomy 2.3GA and is used by iAccess 2 and onwards. This endpoint is used to retrieve the end-user's environment variable, e.g., employee name and number, company info etc.

Chapter 3

Installing iAccess

This section describes the installation process for iAccess. Keep in mind that parts of the installation process (in particular, web server configuration) are specific to the individual installation. As such, this section can only offer general guidelines. In case of doubt, we recommend posting a question on our iAccess Kona Space.

3.1 The iAccess Manifest

To install iAccess, you need an installation of Maconomy and a suitable version of MConfig. The required Maconomy version for a given iAccess FPU (*Flexible Packaging Unit*) is documented in the release notes for each iAccess release. You can also inspect the `manifest.json` file contained in the FPU. In the following `manifest.json` sample, we can see that this iAccess FPU is of version 1.3.4. The dependencies section lists the core Maconomy versions that this version of iAccess is compatible with. In this case, it is only 19sp100. This is the internal version number for the following Maconomy version: 2.3GA.

```
{
  "manifest-version": 2,
  "version": "iAccess for Maconomy 2.0",
  "revision": {
    "major": 2,
    "minor": 0,
    "sp": 0,
    "fix": ""
  },
  "api": "4.0.0",
  "dependencies": {
    "tpu": "19.0.100",
    "apu": "19.0.100"
  }
```

```
}
```

Once the Maconomy system is installed and configured, MConfig can install the iAccess FPU on a web server of choice. We currently support IIS and Apache (see release notes for the specific version requirements for these web server products). On the web server, you should also set up a web site with the installed iAccess `index.html` in the root. Once the MConfig installation has taken place, and the web site and proxy settings have been completed, iAccess is ready.

3.2 Security Considerations

While Deltek recommends the following procedures, ultimately each company is liable for its own security. The landscape evolves quickly, and each firm should continuously take internal measures to ensure its own security.

3.2.1 Regarding the use of HTTPS/TLS

Deltek best practice recommends that you configure web servers to use HTTPS (instead of HTTP). Using HTTPS/TLS encrypts your network traffic, making it difficult for anyone to access the credentials as they are passed to the web server. Using simple HTTP is tantamount to sending confidential information over the wire in clear text.

3.2.2 Address Risk of Clickjacking

To reduce the likelihood of clickjacking, Deltek suggests you follow the OWASP guidelines to defend against clickjacking attacks. Based on the OWASP guideline, you can perform additional steps when configuring your webserver. See Additional Related Procedures for more details.

3.3 Prerequisites

The following are prerequisites to installing iAccess:

- Any of the following Maconomy versions: 2.2.5 GA, 2.3 GA CU3, 2.3.1 CU1, or 2.4 LA1
- MConfig 8.13.1 or later
- Extender 1.6
- RESTful Web Services is enabled in the Coupling Service
- iAccess downloaded from DSM, and iAccess FPU placed in the PUs folder (with the APU and TPU)

- If you are using Apache as the webserver, download the Apache binary package including OpenSSL, and install it from the following link: http://httpd.apache.org/
- Standard extensions are already installed

Additionally, this document assumes that you have already set up an application. For detailed instructions on setting up applications, see the Deltek Maconomy 2.3 Installation Guide.

3.4 MConfig Installation

To begin installation with MConfig, complete the following steps:

Step 1 : In the MConfig Main Window, double-click the application to open. The Application Instance window displays as shown in Figure 14.

Figure 14: The Application Instance window.

Step 2 : Click OSGi products. The OSGi Server Selection screen appears as shown in Figure 15.

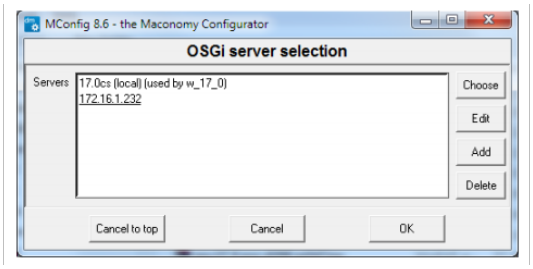

Figure 15: The OSGi Server Selection window.

Step 3 : Select the Coupling Service to update as shown in Figure 16.

Step 4 : Select the Enable RESTful Web Services check box as shown in Figure 17.

Step 5 : Click OK to save, and click OK at the SSL warning to return to the Application Instance window. In the Application Instance window, click Web products as shown in Figure 18.

Note: While you can click OK at the SSL warning, Deltek recommends you follow the steps listed in the warning to ensure the security of your system.

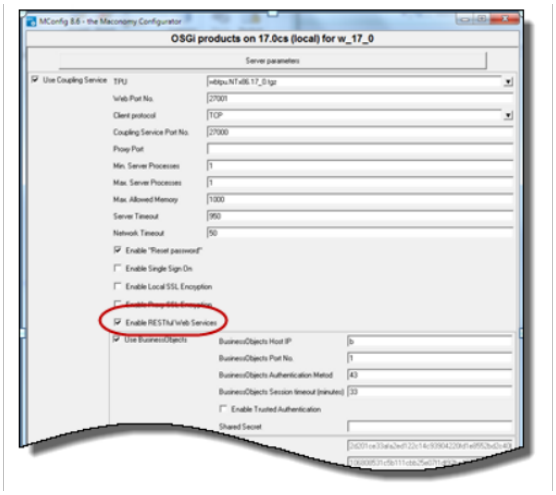

Figure 16: Coupling service selection.

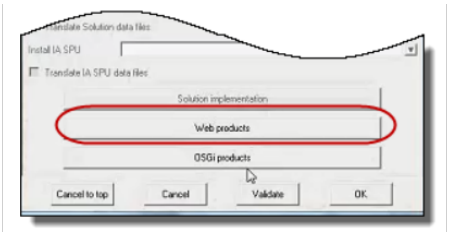

Figure 17: Enable RESTful Web Services.

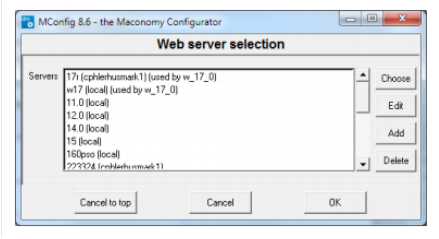

Figure 18: Select web products.

Step 6 : On the Web server selection screen, select the application to update. Select the iAccess check box as shown Figure 19. In the iAccess FPU field, select the relevant FPU from the drop-down list.

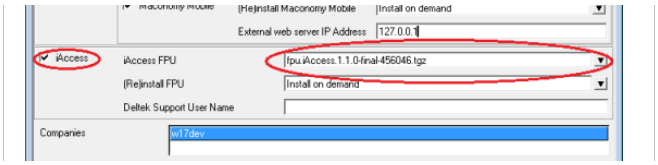

Figure 19: Web products window.

Step 7 : Click Ok a couple of times to return to the main window, and click Next a couple of times, and then click Yes to complete the MConfig installation.

3.5 Create a Website Using IIS

To create a website using IIS, you can enable IIS support automatically using MConfig, or perform the steps manually using IIS Manager.

Enabling IIS support with MConfig automatically completes the steps described under the manual installation. Use MConfig for the initial setup of an iAccess website using IIS. However, modifying the setup later should be done manually using IIS Manager.

3.5.1 Enable IIS Support Automatically Using MConfig

Automatic IIS configuration requires MConfig 8.12.4. Previous version will not perform a correct configuration of IIS due to a shortcoming in MConfig. To enable IIS support using MConfig, follow these steps (See Figure 20):

1. In MConfig, go to the Web Products window.
2. Select the Enable IIS support for iAccess check box and click OK.

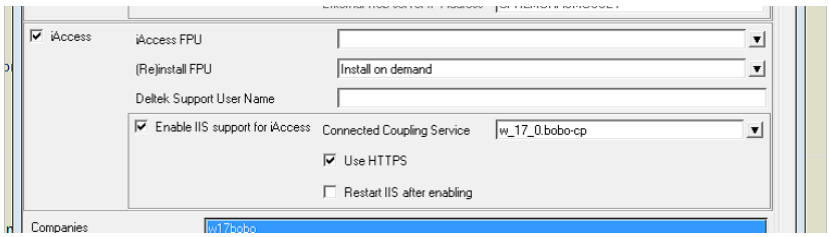

Figure 20: Enable IIS support using MConfig

Note: After you complete the initial installation with MConfig, you should check the setup in IIS Manager and possibly modify parameters, such as Web Server Port Number.

Note: If you enable IIS support automatically, MConfig also updates the IIS web.xml file with a routing rule that ensures login pages and other non-root URLs load properly.

3.5.2 Enable IIS Support Manually Using IIS Manager

After iAccess has been installed on your IIS web server, you can configure it using IIS Manager.

Add the Site

: Connect to your server in the Internet Information Services (IIS) Manager application and setup the iAccess site. The site should have the files shown in Figure 21 as root files.

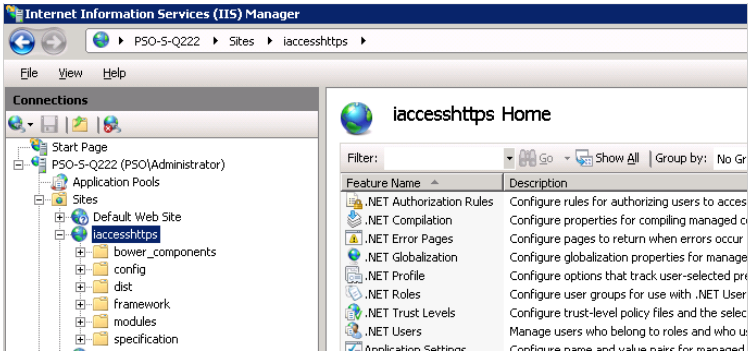

Figure 21: Add Site

Add MIME Types

: Click the ”MIME Types” and ensure that the MIME Types below are defined

```
.json application/json
.woff application/font-woff
.woff2 application/font-woff
```

In IIS 8.0 and up, the .woff extension exists by default but with a different type. Change it to *application/font-woff*.

Proxy Setup

1. Install Microsoft Application Request Routing for IIS ARR.
2. Restart IIS Manager.

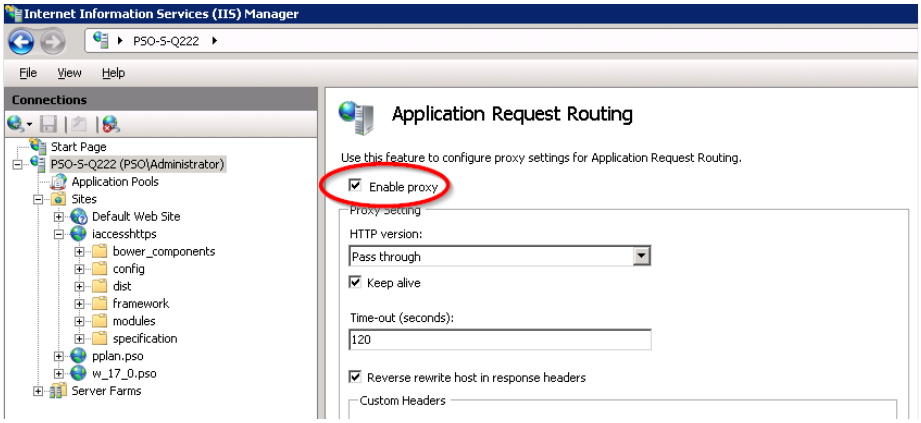

Figure 22: Enable Proxy

3. In the *Application Request Routing* configuration, click *server proxy settings*.
4. Check *Enable proxy* as shown in Figure 22.
5. Open *URL Rewrite* to add proxy rules for the container, configurations and filedrop APIs. Note: This must be done on the local site, not globally as shown in Figure 23.

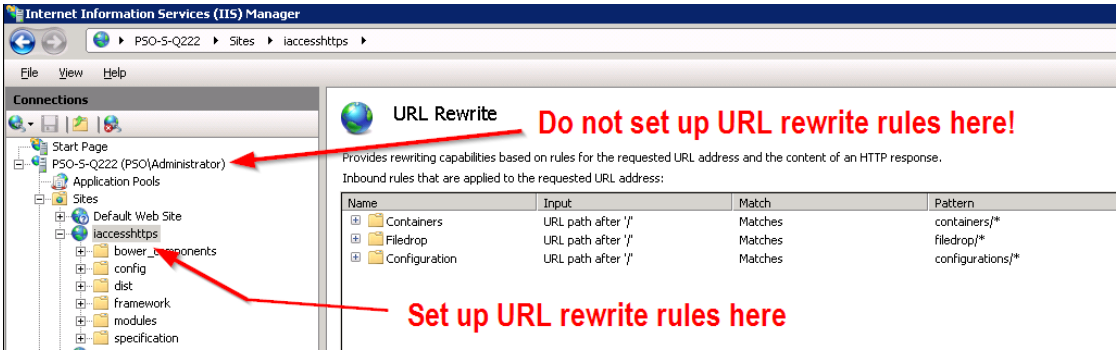

Figure 23: Add Proxy Rules

In IIS 8.0 and up, you will need to install the Web Platform Installer in order to install the ARR plugin.

Edit Routing Rules

To ensure that login pages (and other non-root URLs) load properly, open the IIS web.xml file and add the following rule *before* the other routing rules:

```
<rule name="DeepLinkingSupport" stopProcessing="false">
  <match url=".*" />
    <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
      <add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
```

```
      <add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true"  ↩
    />
      <add input="{REQUEST_FILENAME}" pattern="containers*" negate="true" />
      <add input="{REQUEST_FILENAME}" pattern="filedrop*" negate="true" />
      <add input="{REQUEST_FILENAME}" pattern="configurations*" negate="true ↩
    " />
      <add input="{REQUEST_FILENAME}" pattern="auth*" negate="true" />
      <add input="{REQUEST_FILENAME}" pattern="environment*" negate="true"  ↩
    />
    </conditions>
  <action type="Rewrite" url="/" />
</rule>
```

Set up proxy for Container API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in Figure 24.

Make sure to choose the Coupling Service webport when setting this up, for example, 4111 in Figure 24. The host should be the ip or hostname of the coupling service, not necessarily 127.0.0.1. Here is an overview of the required parameters for the rule.

The rewrite URL *MUST* be HTTP rather than HTTPS. Otherwise the rewriting of response URLs will not work, and iAccess will not be able to make it past the login screen.

Match URL

```
Requested URL: Matches the Pattern
Using: Wilcards
Pattern: containers/*
Ignore case: checked
```

Action

```
Action type: Rewrite
Rewrite URL: http://<coupling-service-host>:<coupling-service-port>/ ↩
    containers/{R:1}
Append query string: checked
```

Set up proxy for Configurations API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in the section on setting up proxy for Containers, specifically with the following parameters:

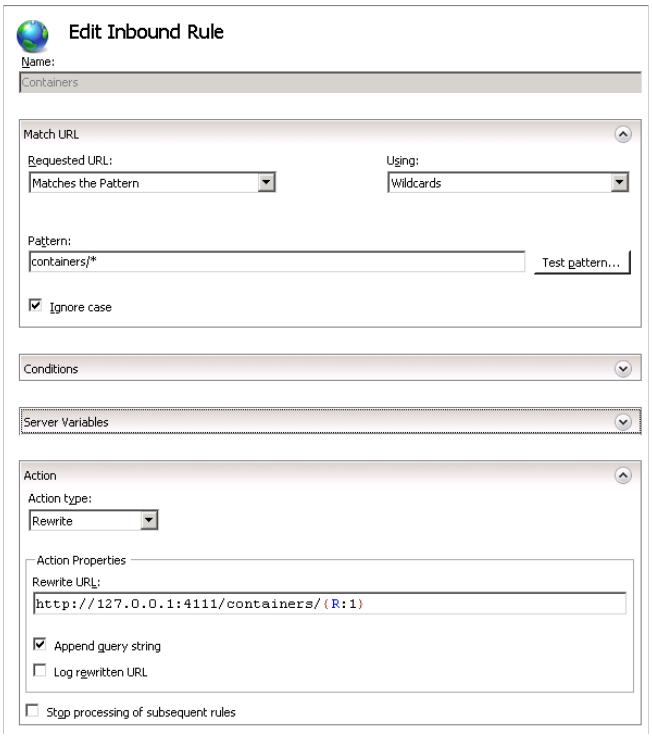

Figure 24: Container API

Match URL

```
Requested URL: Matches the Pattern
Using: Wilcards
Pattern: configurations/*
Ignore case: checked
```

Action

```
Action type: Rewrite
Rewrite URL: http://<coupling-service-host>:<coupling-service-port>/ ↩
    configurations/{R:1}
Append query string: checked
```

Set up proxy for Filedrop API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in the section on setting up proxy for Containers, specifically with the following parameters:

Match URL

```
Requested URL: Matches the Pattern
Using: Wilcards
Pattern: filedrop/*
Ignore case: checked
```

Action

```
Action type: Rewrite
Rewrite URL: http://<coupling-service-host>:<coupling-service-port>/filedrop ↩
    /{R:1}
Append query string: checked
```

Set up proxy for Auth API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in the section on setting up proxy for Containers, specifically with the following parameters:

Match URL

```
Requested URL: Matches the Pattern
Using: Wilcards
Pattern: auth/*
Ignore case: checked
```

Action

```
Action type: Rewrite
Rewrite URL: http://<coupling-service-host>:<coupling-service-port>/auth/{R ↩
    :1}
Append query string: checked
```

Set up proxy for Environment API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in the section on setting up proxy for Containers, specifically with the following parameters:

Match URL

```
Requested URL: Matches the Pattern
Using: Wilcards
Pattern: environment/*
Ignore case: checked
```

Action

```
Action type: Rewrite
Rewrite URL: http://<coupling-service-host>:<coupling-service-port>/ ↩
    environment/{R:1}
Append query string: checked
```

Preserve the Host Header

Open a console with Administrative privileges, and navigate to

```
C:\Windows\System32\inetsrv
```

Enable preserveHostHeader by running the following command:

```
cd C:\Windows\System32\inetsrv
appcmd.exe set config -section:system.webServer/proxy /preserveHostHeader:" ↩
    True" /commit:apphost
```

Note: To preserve the spacing, copy the command and paste it in the command prompt.

Restart the web server.

See *AppCmd* reference for more details.

3.5.3 Set Up HTTPS

Open the Server Variables screen by clicking *View Server Variables...* in the *URL Rewrite* screen.

In the Server Variables screen, click *Add...* and add the variable `HTTP_X_FORWARDED_PROTO` as shown in Figure 27.

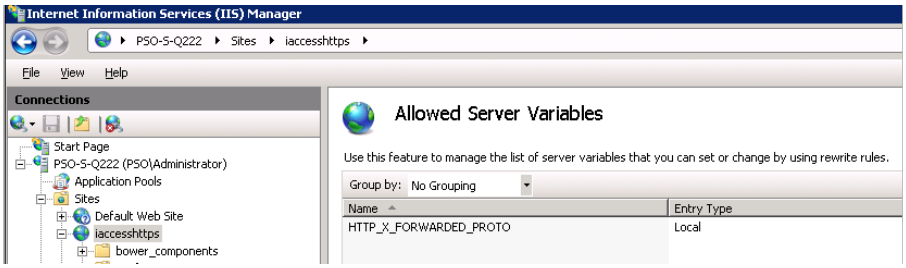

Figure 27: Add Server Variables

In the URL rewrite rules (both containers, configurations, filedrop, auth, and environment) that proxies the web service, set the server variable `HTTP_X_FORWARDED_PROTO` to `https` as shown in Figure 28.

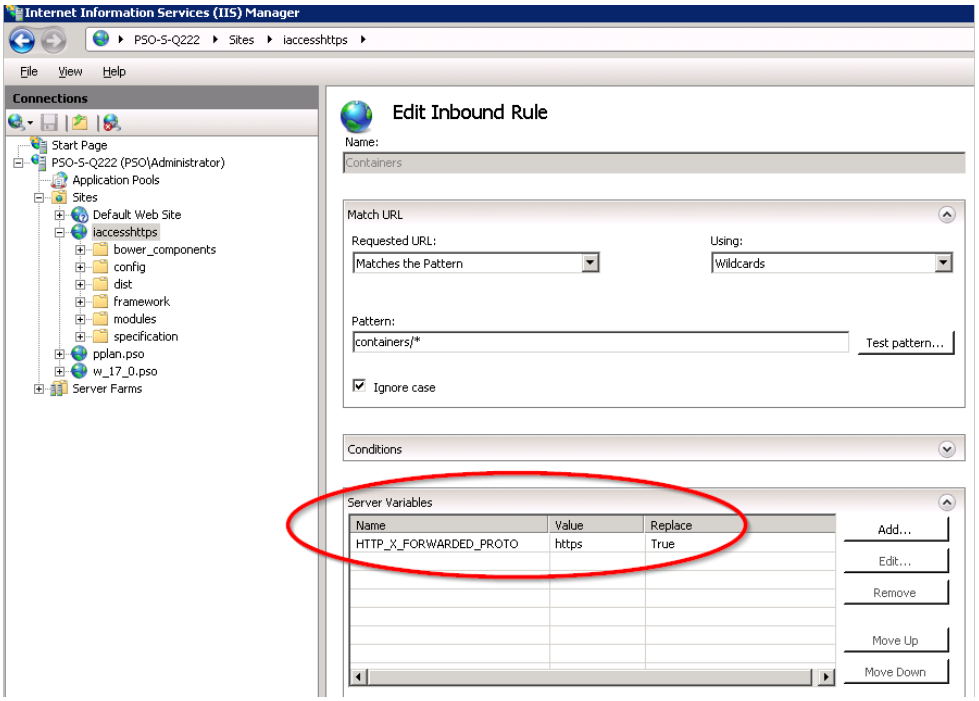

Figure 28: Setting up HTTPS

Restart the webserver.

Note: It is not possible to run both HTTP and HTTPS on the same IIS site.

3.6 Create a Website Using Apache

Here is a short guide to setting up iAccess on Apache (2.2 and 2.4)

3.6.1 Download Apache

Download the Apache 2.2 binary package including OpenSSL. Install it.

In `httpd.conf`, comment in the following modules:

```
LoadModule headers_module modules/mod_headers.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule rewrite_module modules/mod_rewrite.so
```

Comment in the inclusion: `Include conf/extra/httpd-vhosts.conf`

Comment out `#Listen 80` (we will use the `httpd-vhosts.conf` file instead)

3.6.2 Enable compression

If using Apache 2.2, comment in the following module: `LoadModule deflate_module modules/mod_deflate.so`

If using Apache 2.4, use this module: `LoadModule filter_module modules/mod_filter.so`

Configure the compression:

```
<IfModule deflate_module>
  AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css text/ ↩
    javascript   application/javascript
  SetOutputFilter DEFLATE
  DeflateCompressionLevel 5
</IfModule>
```

See Apache Deflation Module for more information.

3.6.3 Setup without SSL

Here is a template for setting up a virtual host that serves iAccess without SSL. Copy contents into the `httpd-vhosts.conf` file, and replace the variables with the desired values.

The `<server-name>` and `<port>` is the host name and port number used to expose iAccess externally. The `<iAccess-installation-directory>` is the folder containing iAccess's `index.html` file, for example, `C:/Maconomy/Webservers/iaccess/app`.

```
Listen <port>
<VirtualHost *:<port>>
    ServerName <server-name>

    # Server iAccess files from installation directory
    DocumentRoot "<iAccess-installation-directory>"

    <Directory <iAccess-installation-directory>>
       Order deny,allow
       Allow from all
       AllowOverride All
       Require all granted
    </Directory>

    <Proxy *>
       Order deny,allow
       Allow from all
    </Proxy>

    ProxyRequests      Off
    ProxyPreserveHost  On

    # Proxy the web services from the coupling service
    ProxyPass  /containers                  http://<coupling-service-host>:< ↩
    coupling-service-web-port>/containers                retry=0
    ProxyPass  /filedrop                    http://<coupling-service-host>:< ↩
    coupling-service-web-port>/filedrop                  retry=0
    ProxyPass  /configurations              http://<coupling-service-host>:< ↩
    coupling-service-web-port>/configurations            retry=0
    ProxyPass  /auth                        http://<coupling-service-host>:< ↩
    coupling-service-web-port>/auth                      retry=0
    ProxyPass  /environment                 http://<coupling-service-host>:< ↩
    coupling-service-web-port>/environment               retry=0
</VirtualHost>
```

Here is an example using the preceding template:

```
Listen 8090
<VirtualHost *:8090>
    ServerName techwebproject

    # Server iAccess files from installation directory
    DocumentRoot "C:/Maconomy/Webservers/iaccess/app"

    <Directory C:/Maconomy/Webservers/iaccess/app>
       Order deny,allow
```

```
      Allow from all
      AllowOverride All
      Require all granted
   </Directory>

   <Proxy *>
      Order deny,allow
      Allow from all
   </Proxy>

   ProxyRequests      Off
   ProxyPreserveHost  On

   # Proxy the web services from the coupling service
   ProxyPass  /containers               http://localhost:8085/containers    ↩
              retry=0
   ProxyPass  /filedrop                 http://localhost:8085/filedrop      ↩
              retry=0
   ProxyPass  /configurations           http://localhost:8085/ ↩
   configurations          retry=0
   ProxyPass  /auth                     http://localhost:8085/auth          ↩
              retry=0
   ProxyPass  /environment              http://localhost:8085/environment   ↩
              retry=0
</VirtualHost>
```

3.6.4 Setup with SSL

Here is a template for setting up a virtual host that serves iAccess *with* SSL. Copy contents into the `httpd-vhosts.conf` file, and replace the variables with the desired values.

```
Listen <port>
<VirtualHost *:<port>>
    ServerName <server-name>

    # Server iAccess files from installation directory
    DocumentRoot "<iAccess-installation-directory>"

    <Directory <iAccess-installation-directory>>
       Order deny,allow
       Allow from all
       AllowOverride All
       Require all granted
    </Directory>

    <Proxy *>
       Order deny,allow
```

```
      Allow from all
    </Proxy>

    ProxyRequests      Off
    ProxyPreserveHost  On
    # Signal to the coupling service that the originating protocol is HTTPS
    RequestHeader set X-Forwarded-Proto "https"

    # Proxy the web services from the coupling service
    ProxyPass  /containers                  http://<coupling-service-host>:< ↩
    coupling-service-web-port>/containers                retry=0
    ProxyPass  /filedrop                    http://<coupling-service-host>:< ↩
    coupling-service-web-port>/filedrop                  retry=0
    ProxyPass  /configurations              http://<coupling-service-host>:< ↩
    coupling-service-web-port>/configurations            retry=0
    ProxyPass  /auth                        http://<coupling-service-host>:< ↩
    coupling-service-web-port>/auth                      retry=0
    ProxyPass  /environment                 http://<coupling-service-host>:< ↩
    coupling-service-web-port>/environment               retry=0

    # Set up this virtual host to use SSL
    SSLEngine              On
    SSLProxyEngine         On
    SSLCertificateFile     <crt-file-location>
    SSLCertificateKeyFile  <key-file-location>
</VirtualHost>
```

Here is an example using the preceding template:

```
Listen 443
<VirtualHost *:443>
    ServerName techwebproject

    # Server iAccess files from installation directory
    DocumentRoot "C:/Maconomy/Webservers/iaccess/app"

    <Directory C:/Maconomy/Webservers/iaccess/app>
       Order deny,allow
       Allow from all
       AllowOverride All
       Require all granted
    </Directory>

    <Proxy *>
       Order deny,allow
       Allow from all
    </Proxy>

    ProxyRequests      Off
    ProxyPreserveHost  On
```

```
    # Signal to the coupling service that the originating protocol is HTTPS
    RequestHeader set X-Forwarded-Proto "https"

    # Proxy the web services from the coupling service
    ProxyPass  /containers              http://localhost:8085/containers    ↩
               retry=0
    ProxyPass  /filedrop                http://localhost:8085/filedrop      ↩
               retry=0
    ProxyPass  /configurations          http://localhost:8085/ ↩
    configurations          retry=0
    ProxyPass  /auth                    http://localhost:8085/auth          ↩
               retry=0
    ProxyPass  /environment             http://localhost:8085/environment   ↩
               retry=0

    # Set up this virtual host to use SSL
    SSLEngine              On
    SSLProxyEngine         On
    SSLCertificateFile     c:/sslkeys/server.crt
    SSLCertificateKeyFile  c:/sslkeys/server.key
</VirtualHost>
```

3.6.5 Edit Routing Rules

For more information, refer to Edit Routing Rules under the Enable IIS Support Manually Using IIS Manager section.

3.6.6 Verifying the setup

A quick way to verify the setup is to execute a cURL command that makes a web service call. For example:

```
curl -k -u'Administrator:123456' https://techwebproject/containers/v1/ ↩
    w17pso2/timesheets
```

Where `Administrator` is the username, `123456` is the password and `w17pso2` is the shortname.

3.7 Domain Login and Single Sign On

The domain login functionality in iAccess is based on Kerberos service tickets obtained through the SPNEGO authentication protocol. This protocol allows direct Single Sign On (SSO) when the user is running iAccess while already authenticated against the domain (that is, logged in to their computer via a domain account).

If the user is not authenticated against the domain, the browser typically prompts for domain credentials. Click **Cancel** in the browser login window and use the iAccess domain login page.

3.7.1 Browser Setup for Single Sign On

Refer to the instructions in this section to set up Single Sign On (SSO) for various browsers.

SSO Setup for Internet Explorer

For Internet Explorer (IE), you may need to add the iAccess server address to the Local intranet zone if it is not already in this zone, as IE does not permit Kerberos-based SSO for websites in the Internet zone.

More details are available in the "Client Side-Internet Explorer" section of the following Microsoft article about security zones in Internet Explorer:

`https://msdn.microsoft.com/en-us/library/ms995329.aspx`

SSO Setup for Chrome

You can choose one of two options:

- If using Windows, you can perform the setup required for Internet Explorer. Chrome can replicate the setup for IE.
- To configure Chrome to work with SSO using Kerberos authentication, follow the steps in the "Set Chrome policies for devices" guide (https://support.google.com/chrome/a/answer/

The configurations should be done by IT administrators who want to set Chrome policies on their corporate-managed devices. The templates contain hundreds of available policies that can be set, but you should only focus on two of these, namely:

AuthNegotiateDelegateWhitelist (http://www.chromium.org/administrators/policy-list- 3#AuthNegotiateDelegateWhitelist)

and

AuthServerWhitelist (http://www.chromium.org/administrators/policy-list-3#AuthServerWhitelist).

The properties should be set to the domain you want to authenticate against, such as:

`*. example.com .`

SSO Setup for Chrome on Windows

After following the preceding guide from Google, you can set **AuthNegotiateDelegateWhitelist** and **AuthServerWhitelist** as follows:

1. Navigate to Administrative Templates » Classic Administration Templates (ADM) » Google » Google Chrome » Policies for HTTP Authentication.
2. Click “Kerberos delegation server whitelist”.
3. Click Enabled.
4. In the Input field, enter the domain you want to authenticate against, such as “*. example.com ”.
5. Click Apply.
6. Click on “Authentication server whitelist”.
7. Click Enabled.
8. In the input field, enter the domain you want to authenticate against, such as “*. example.com ”.
9. Click Apply.
10. Open Chrome.
11. Check the values by navigating to the URL:

    `chrome://policy`

SSO Setup for Chrome on Mac

After following the preceding guide from Google, you should also read the Mac Quick Start guide from Google at:

`http://www.chromium.org/administrators/mac-quick-start`

If the “Workgroup Manager from Apple” is not available for your version of OS X, then you can set **AuthNegotiateDelegateWhitelist** and **AuthServerWhitelist** using one of two recipes.

By creating a com.google.Chrome.plist file:

1. Create com.google.Chrome.plist file with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"     "http://www. ↩
    apple.com/DTDs/PropertyList-1.0.dtd">
```

```
<plist version="1.0">
  <dict>
    <key>AuthNegotiateDelegateWhitelist</key>
    <string>*.example.com</string>
    <key>AuthNegotiateDelegateWhitelist</key>
    <string>*.example.com</string>
  </dict>
</plist>
```

2. Set the two string attributes to the domain you want to authenticate against.
3. Convert the com.google.Chrome.plist to the binary format by running the following command from the Terminal:

   ```
   plutil -convert binary1 com.google.Chrome.plist
   ```

4. Copy the file to "/Library/Managed Preferences/" by running the following command from the Terminal:

   ```
   sudo -s
   cp com.google.Chrome.plist /Library/Managed Preferences/<username>
   ```

5. Open Chrome.
6. Check the values by navigating to the following URL:

   ```
   chrome://policy
   ```

By using the "defaults" command:

1. Run the following commands:

   ```
   defaults write com.google.Chrome AuthServerWhitelist *.example.com
   defaults write com.google.Chrome AuthNegotiateDelegateWhitelist *. ↩
       example.com
   ```

2. Open Chrome.
3. Check the values by navigating to the following URL:

   ```
   chrome://policy
   ```

Or by using the following command:

```
defaults read com.google.Chrome
```

Note that using the "defaults" command only sets the "AuthServerWhitelist" and "AuthNegotiateDelegateWhitelist" Chrome properties for the current user.

SSO Setup for Safari

No setup is needed.

SSO Setup for iOS

Follow the steps in the following guide:

`https://samuelyates.wordpress.com/2013/10/11/kerberos-single-sign-on-in-ios-7/`

Remember to put your own servername on the URLPrefixMatches field. As the name implies, this has to contain a URL prefix. This could be "https://myserver.example.com:8080", so basically this will be set to the base server URL including an optional port number.

You can use Apple Configurator 2 to install the profile on a number of iPads:

`https://itunes.apple.com/us/app/apple-configurator-2/id1037126344`

SSO Setup for Firefox

Complete the following steps:

1. In the location bar, type: about:config.

This brings up the configuration page.

2. In the Filter:box, type: negotiate.

This restricts the listing to the configuration options you need.

3. Edit network.negotiate-auth.trusted-uris to the domain against which you want to authenticate. For instance: ".".example.com

3.8 Domain Controller Setup

The SPNEGO authentication protocol works by assuming the presence of a specific Service Principal Name (SPN) on the domain controller:

`HTTP/name.domain or HTTP/name`

where name and domain are the web server DSN name and domain respectively, as seen from the user's computer.

For example, if the user is opening iAccess using the internet address https://some-server.some-domain.com, then the browser expects one of the following SPNs to be present on the domain controller:

`HTTP/some-server or HTTP/some-server.some-domain.com.`

3.8.1 SPN Setup

It is a task for the domain administrator to ensure that these SPNs are created and associated with the existing domain account used for Maconomy SSO.

SSO with Active Directory

For Active Directory, associating SPNs with the existing domain account is done with the 'setspn' command.

To associate SPN with an existing domain account, complete the following step:

On a command line enter the following:

```
setspn -A HTTP/name account
setspn -A HTTP/name.domain account
```

where account is the name of the domain account used for Maconomy SSO.

Special Instructions for SPN Conflicts

If iAccess is installed on a web server that already hosts other web applications with SNEGO authentication, this causes a conflict on the SPN, as an SPN can only be associated with one domain account.

To resolve the issue, either install only one web application on each web server, or create multiple local DNS names for the web server, so that each web application can be accessed through different addresses and will map to different SPNs.

3.9 Maconomy Server Setup

Please refer to the Single Sign On with Kerberos section in the Deltek Maconomy System Administrator Guide.

3.10 Additional Related Procedures

3.10.1 Configure Web Server to Reduce Risk of Clickjacking

You can reduce the risk of clickjacking by performing an additional step when configuring your web server. This step applies to both Apache and IIS.

To configure your web server and reduce the risk of clickjacking, complete the following step:

Configure your web server to always reply with the following response headers:

```
Content-Security-Policy: frame-ancestors self
X-Frame-Options: SAMEORIGIN
```

These headers are then added to all responses.

3.10.2 Downloading Deltek Products using the Deltek Software Manager

You can use the Deltek Software Manager (DSM) to download complete Deltek products, hot fixes, and sub-releases. You can access DSM directly or through the Deltek Customer Care Connect site.

When you access DSM directly, you will be prompted to log on before you can access the application. If you access DSM from within the Deltek Customer Care site, you do not have to log on since you are already logged in to the Customer Care site.

Accessing DSM Directly

To access Deltek Software Manager directly, complete the following steps:

1. Launch Deltek Software Manager by taking one of the following actions:

   Click here: http://www.deltek.com/ On your desktop, click Start » Programs » Deltek » Maconomy iAccess » Deltek Software Manager.

2. In the Deltek Software Manager logon dialog box, enter your Deltek Customer Care User ID and Password, and click Logon.

3. To select the folder where you want to download Deltek products, click Settings above the right pane of Deltek Software Manager.

   Note: When you log on for the first time, Deltek Software Manager asks you to select a default folder where Deltek products are to be downloaded.

4. Use the Settings dialog box to specify the folder where you want to download Deltek products, and click OK.

   Note: You can change this folder anytime in the Settings dialog box.

5. In the left pane of Deltek Software Manager, expand the Deltek product that you want to download, if it is not already expanded.

   Note: If you clicked the link in step 1 to access DSM, the application automatically selects Maconomy iAccess for you.

6. Select the product type that you want to download. Your options are Complete, HotFix, and Sub-Release.

7. In the table, select the check box that corresponds to the Deltek product that you want to download. The right pane displays a message stating that the product has been added to the download queue. To view the items in the download queue, click View Download Queue at the bottom of the left pane.

8. Click Download at the bottom of the left pane. Deltek Software Manager downloads the product to the folder that you selected.

Accessing DSM from within the Customer Care Connect Site

To access Deltek Software Manager from within the Customer Care Connect site, complete the following steps:

1. In your Web browser, go to http://support.deltek.com.

2. Enter your Customer Care Connect Username and Password, and click Log In.

3. When the Customer Care Connect site displays, click the Product Downloads tab. You are automatically logged into Deltek Software Manager.

4. To select the folder where you want to download Deltek products, click Settings above the right pane of Deltek Software Manager.

   Note: When you log on for the first time, Deltek Software Manager asks you to select a default folder where Deltek products are to be downloaded.

5. Use the Settings dialog box to specify the folder where you want to download Deltek products, and click OK.

   Note: You can change this folder anytime in the Settings dialog box.

6. In the left pane of Deltek Software Manager, expand the Deltek product that you want to download, if it is not already expanded.

7. Select the product type that you want to download. Your options are Complete, HotFix, and Sub-Release.

8. In the table, select the check box that corresponds to the Deltek product that you want to download. The right pane displays a message stating that the product has been added to the download queue.

   Note: To view the items in the download queue, click View Download Queue at the bottom of the left pane.

9. Click Download at the bottom of the left pane. Deltek Software Manager downloads the product to the folder that you selected.

DSM Documentation and Troubleshooting

To view the online help for Deltek Software Manager, navigate to:

`https://dsm.deltek.com/DeltekSoftwareManager/Help/`

To view a tutorial on how to use Deltek Software Manager, navigate to:

```
https://dsm.deltek.com/DeltekSoftwareManager/Tutorial/PubData/Engine/Default ↩
    .htm?https%3A%2F%2Fdsm.deltek.com%2FDeltekSoftwareManager%2FTutorial%2 ↩
    FPubData%2F
```

To view more information on troubleshooting Deltek Software Manager, navigate to:

`https://deltek.custhelp.com/app/answers/detail/a_id/52469`

Note: The preceding troubleshooting link only works if you are logged in to Deltek Customer Care Connect.

Chapter 4

Extending iAccess

The Extension model for iAccess 2 has changed significantly. We will provide further information about the new facilities in an upcoming version of the Install Guide. For now, please direct any questions to the Engineering team through our “iAccess for Maconomy” Kona space (https://www.kona.com/#!/projects/129727).

Chapter 5

Miscellaneous

The following section contains a migration guide which describes how to migrate a customized iAccess installation. We also include a troubleshooting guide with a few tips about how to overcome typical installation issues.

5.1 Migration Guide

The following sections describes the steps needed for migrating from one specific version of iAccess to another. If the version that you are currently using is not mentioned here, or if you have received a special release or hotfix, please get in touch with the iAccess development team for further instructions.

5.1.1 From 1.x to 2.0

You need to redo all extensions from scratch. We are not delivering a migration tool at this point. Deltek recommends that you inspect the standard extensions delivered with iAccess 2.0, and then ask concrete questions in our “iAccess for Maconomy” Kona space (https://www.kona.com/#!/projects/129727). Engineering is actively monitoring this forum to ease the transition.

Please observe that for 2.0, two new rewrite rules are added for the two new REST endpoints that we rely on, specifically “auth” and “environment”. On IIS, an additional rewrite rule is required as described in the “Edit Routing Rules” section. Also, you need to configure “woff2” as a supported MIME type.

5.1.2 From 1.2.x and 1.3.0-3 to 1.3.4

For upgrades from the 1.2.x series, perform the steps outlined in the following sections, and then proceed to the step given here. When migrating from 1.3.0-3 versions to 1.3.4, make sure you update all JSON references. You no longer need to include an iAccess namespace in the names of specification files. This means you should remove the `iaccess:`-prefix from JSON references. For example, change:

```
"screens": {
  "dm.dailytimesheets": {
    "$ref": "iaccess:dailytimesheets"
  }
```

to

```
"screens": {
  "dm.dailytimesheets": {
    "$ref": "dailytimesheets"
  }
```

5.1.3 From 1.2.0 and 1.2.1 to 1.2.2 and 1.3.0

Some of our core terminology has changed, as outlined in the following table. This means that you need to update the following keys in your configuration:

User Interface Concept	1.2.0 and 1.2.1 API	1.2.2 and 1.3 API
Default View	`defaultScreen`	`defaultView`
Views	`screens`	`views`
Leftnav	`sidebar`	`leftnav`

Since this is a *breaking change*, the API version has also changed. Configurations should state that they now rely on version `2.0.0` rather than `1.2.0`.

In 1.3, we added support for the Additional Table Fields extension point in expense and mileage sheets. This means you must merge the specifications from these views with the new defaults from the iAccess 1.3 FPU. You can use the Maconomy Extender to assist in this process.

We also added some new parts to the configuration. Integrate these changes by importing the latest specifications from a new FPU and merging these with existing customizations. Specifically, we introduced a `documentationUrl` under the `configuration` section in `application.json`. In this section, we also introduced references to two new specification

files: `authentication.json` and `usagetracking.json`. Use the Maconomy Extender to add these files.

When migrating to `1.2.2` or to `1.3.0`, remove the preferences in the following listing from the `preferences.json` file. The defaults have changed and are not valid anymore. Customize the preferences as described in the configuration section.

```
"dateFormat": {
 "short": "M/d/yyyy"
},
"decimalSymbol": ",",
"digitGroupingSystem": ".",
"minutesThreshold": 10
```

5.1.4 From 1.1.x to 1.2.x

The major difference between the 1.x versions and 1.2.x is the introduction of the extensibility model. The table in the next section describes which configuration options from 1.2.x replace deprecated configuration options from 1.1.x.

Configuration of Leftnav

In version 1.1.x, you could configure which views were accessible via the leftnav by changing the `this.sidebarItems` array in `config.js`. This was done after installation on each individual web server.

In version 1.2.x, this kind of configuration is now a part of the central view configuration, and managed via the Maconomy Extender. You can access the configuration of each view by following the links from `application.json` file. To show/hide a particular view, set the `enabled` attribute to either `true` or `false`.

```
{
  "name": "dm.dailytimesheets",
  "enabled": "true",
  ...
}
```

The mapping of view names between version 1.1.x and 1.2.x can be found in the following table:

1.1.x View Name	1.2.x View Name
inside.timesheets	dm.weeklytimesheets
inside.dailytimesheets	dm.dailytimesheets
inside.expensesheets.edit	dm.expensesheets

1.1.x View Name	1.2.x View Name
inside.mileagesheets.edit	dm.mileagesheets
inside.jobfavorites	dm.favoritemgmt
inside.absence.tabs	dm.absencemgmt

In version 1.1.x, you could specify the default leftnav tab in `config.js` with the `defaultSidebarItem` property. In version 1.2.x, you specify the default leftnav in the beginning of the `application.json` configuration as shown in the following example:

```
{
  "api": "1.2.0",
  "defaultScreen": "dm.weeklytimesheets",
  "screens": ...
}
```

Configuration of the Weekly Time Sheets View

In version 1.1.x, you could configure two properties of the time sheets' views: daily descriptions, and overtime specification.

In weekly timesheets, you could enable or disable daily descriptions in `config.js` by setting the `isDailyDescriptionsEnabled` property to either true or false. In version 1.2.x, you achieve this configuration by using the extension point `dm.additionalTableFields` described in a previous section.

Finally, in version 1.1.x, you could show or hide the overtime specification in weekly time sheets. In version 1.2.x, you achieve this by adding the `overtimetype` field to the table using the extension point `dm.additionalTableFields` described in a previous section.

Localization

In version 1.1.x, you localized a subset of the terms (for example, error messages) by placing a custom iAccess dictionary in the `i18n` folder on each web server. The 1.1.x version was only released with dictionaries for English and Danish. In version 1.2.x, all localization takes place on the Maconomy server through the existing localization engine. You customize translations by editing the traditional Maconomy dictionaries on the Maconomy server.

5.2 Troubleshooting Guide

Solutions to common installation issues are found in the Installing iAccess section. If your issue/problem is not listed there, the following section provides some additional clues to solve common issues. If you still cannot find a solution to your specific problem, please post a conversation in the *iAccess for Maconomy* Kona space or raise a support case through Customer Care to get your concrete issue resolved.

A piece of general advice for technical consultants: Always take a look at the requests that the browser issues when you are getting installation and/or network problems. In particular, the AJAX requests and error responses are often useful for uncovering installation and configuration errors. Figure 35 shows Developer Tools in Chrome where the Network Tab can be a very powerful tool to uncover installation and network problems.

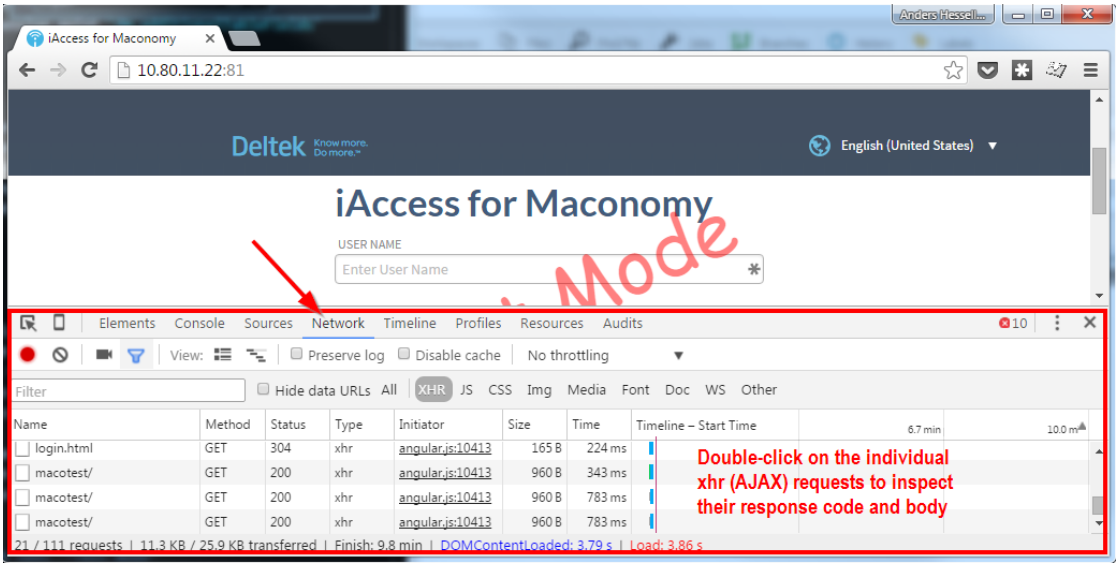

Figure 35: Use the Network Tab in your browser's Developer tools to debug failing requests and login problems.

"*Incompatible API versions...*"-Error

The iAccess specification format deployed on the Maconomy server is not compatible with the installed version of iAccess. This error usually occurs because either the `application.json` specification or the iAccess installed on a given web server have not been updated as part of a system upgrade. If the lowest number in the error message is the *required API version*, then you need to upgrade the iAccess version installed on the given web server. This requires the use of MConfig.

If the lowest number is the *loaded specification API version*, then update the specification deployed to the Maconomy server. This requires the Maconomy Extender.

"*Bad Request: Unable to connect to 'configurations' endpoint...*"-Error

When moving from version 1.1.x to 1.2.x, iAccess becomes dependent on a new webservice called *configurations*. This web service has to be available through the proxy configuration on the web server. This is similar to how the *containers* and *filedrop* web services are setup. See the Installing iAccess section for details.

Even if you have properly configured the *configurations* endpoint, you may still get the error on certain IIS installations. The problem can then be that some IIS installations do not allow the colon : character in URLs. To solve this, allow the colon : in the `web.config` file in the root of your web server [1]. Use its unicode encoded format `%u003a` in the configuration.

```
<configuration>
  <system.web>
  <!-- Default <,>,*,%,&,:,\,?
            or %u003c,%u003e,%u002a,%u0025,%u0026,%u003a,%u005c,%u003f -->
    <httpRuntime
       requestPathInvalidCharacters="%u003c,%u003e,%u002a,%u0025,%u0026,% ↩
    u005c,%u003f" />
  </system.web>
  <system.webServer>
    ...
  </system.webServer>
</configuration>
```

"*A%20Network%20Error%20Occurred*"-Window Opens

This error occurs when HTTPS has been partially or incorrectly configured on the web server. Double-check that the web server is configured according to the steps in the Installing iAccess section. This includes checking that the OSGi products in MConfig are configured correctly, and that HTTPS forwarding rules are set up on the web server.

Chapter 6

Figures

(25) Configurations API
(26) Filedrop API
(27) Add Server Variables
(28) Setting up HTTPS
(29) iAccess in *Test Mode*
(30) Maconomy Extender 1.6 with a sample Maconomy Extender project
(31) Import iAccess specification from an FPU
(32) Locate the right iAccess FPU
(33) *Web*-folder with iAccess specifications
(34) Commit and push iAccess specifications
(35) Use the Network Tab in your browser's Developer tools to debug failing requests and gin problems.

Bibliography

[1] How to make IIS allow colon sign in request url, February 2015. URL http://www.avantec.se/howto-make-iis-allow-colon-sign-in-request-url/.

[2] *Maconomy RESTful Web Services—Programmer's Guide*. Deltek Inc., September 2015.