Reverse Proxy for Maconomy

Setup Guide
Version 1.3, November 30, 2018

Edited by

Andrew Barrett
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 November 2018.

Contents

Revision History 1

1 Introduction 3

2 Coupling Service and Reverse Proxy Architecture 5
2.1 Technical Architecture . . . 6
2.1.1 RESTful Web Service API (iAccess & Integrations) . . . 7
2.1.2 OnePort (Workspace Client) . . . 7

3 Installing RESTful Web Service API 9
3.1 Security Considerations . . . 9
3.1.1 Regarding the use of HTTPS/TLS . . . 9
3.1.2 Address Risk of Clickjacking . . . 10
3.2 Prerequisites . . . 10
3.3 MConfig Installation . . . 10

4 Installing OnePort 13
4.1 Security Considerations . . . 13
4.1.1 Regarding the use of HTTPS/TLS . . . 13
4.1.2 Address Risk of Clickjacking . . . 13
4.2 Prerequisites . . . 14
4.3 MConfig Installation . . . 14

5 Create a Website Using IIS 17
5.1 Enable IIS Support Automatically Using MConfig . . . 17
5.1.1 Proxy Setup . . . 18
5.1.2 MConfig Setup . . . 18
5.2 Enable IIS Support Manually Using IIS Manager . . . 22
5.2.1 Add the Site . . . 22
5.2.2 Add MIME Types . . . 23
5.2.3 Proxy Setup . . . 23
5.2.4 Edit Routing Rules . . . 24
5.2.5 Set up proxy for Containers API . . . 25

5.2.6 Set up proxy for Configurations API . . . 26
5.2.7 Set up proxy for Filedrop API . . . 27
5.2.8 Set up proxy for Auth API . . . 28
5.2.9 Set up proxy for Environment API . . . 29
5.2.10 Set up proxy for Handshake API . . . 29
5.2.11 Set up proxy for Web Sockets API . . . 30
5.2.12 Set up proxy for Analyzer API . . . 30
5.2.13 Preserve the Host Header . . . 31
5.2.14 Set Up HTTPS . . . 31
5.2.15 Example web.config . . . 33

6 Create a Website Using Apache 37
6.1 Download Apache . . . 37
6.2 Enable Apache Support Using MConfig . . . 37
6.3 Enable Apache Support Manually . . . 38
6.4 Enable Compression . . . 38
6.5 Setup with SSL . . . 39
6.6 Edit Routing Rules . . . 42

7 Verifying the Reverse Proxy Setup 43

8 Additional Configuration Procedures 45
8.1 Conditionally Setting Request Headers . . . 45
8.1.1 IIS . . . 45
8.1.2 Apache . . . 46
8.2 Configure Web Server to Reduce Risk of Clickjacking . . . 47
8.2.1 IIS . . . 47
8.2.2 Apache . . . 48
8.3 Domain Login and Single Sign On . . . 48
8.3.1 IIS . . . 48
8.3.2 Apache . . . 49
8.4 OnePort: WebSocket Setup IIS . . . 49
8.5 OnePort: Timeout and Keep-Alive Settings for IIS . . . 50
8.6 OnePort: Set Web Socket Endpoint . . . 51
8.7 OnePort: Set Client Handshake Web Socket URI . . . 52
8.8 OnePort: Force Protocol, Host and Port . . . 52
8.9 OnePort: Required for Maconomy 2.3.3 and 2.4.0 . . . 53
8.9.1 MConfig Details . . . 53
8.9.2 SSL/TLS Details . . . 54

9 Troubleshooting Guide 57

10 Downloading Deltek Products using the Deltek Software Manager 61
10.1 Accessing DSM Directly . . . 61

10.2 Accessing DSM from within the Customer Care Connect Site 62
10.3 DSM Documentation and Troubleshooting 63

11 Figures **65**

Bibliography **67**

Revision History

Date	Author	Notes
2017-09-22	AB	First version of the Revere Proxy Setup Guide.
2018-01-11	AB	Minor verbiage changes. Added note about People Planner.
2018-04-20	AB	Added Details to Coupling Service and Reverse Proxy Architecture to delineate RestAPI and OnePort.
2018-11-30	AB	Added new MConfig feature to setup Reverse Proxies for both Apache and IIS. Minor error corrections.

Chapter 1

Introduction

The following document serves as setup guide for a Web Server Reverse Proxy for a Maconomy Coupling Service using either Internet Information Services (IIS) or Apache. The target audience is technical consultants and partners that need to install and configure access to the Maconomy RESTful Web Service API or configure OnePort for the Workspace Client. Other Reverse Proxy software could be used instead of IIS or Apache. You can use this guide as an inspiration to determine how those products should be configured.

The Maconomy RESTful Web Service API provides a standardized access to a Maconomy instance via the Coupling Service. Several Deltek products take advantage of this RESTful Web Service API including iAccess, People Planner and TrafficLive. The Maconomy RESTful Web Service API can also be used to integrate with third party products and other custom software.

OnePort for Workspace Client allows a single connection to the Coupling Service for communication with a Maconomy server instance. OnePort takes advantage of Web Sockets to provide duplex communication. A Reverse Proxy in front of the Coupling Service provides SSL Termination.

You should note that this guide is not a simple step-by-step implementation guide. All details to complete a setup are contained in this guide, but it is the responsibility of a technical consultant to determine which Coupling Service endpoints are required for a specific installation. It is anticipated that most installations will require both RestAPI and OnePort. But, some installations might require only RestAPI while others require only OnePort.

We also refer the reader to the Kona spaces, *Maconomy RESTful Web Services* and *iAccess for Maconomy*, where Product Management and the Development Team will answer questions and discuss feature requests for the products.

Chapter 2

Coupling Service and Reverse Proxy Architecture

The Maconomy coupling Service provides access to a Maconomy server instance. The Coupling Service provides two types of access: a RESTful Web Service API and Web Sockets.

Briefly described, the Maconomy RESTful Web Service API uses Representational State Transfer (REST) to provide stateless system access to a Maconomy instance. REST is one type of a Web Service. Other examples of Web Services are SOAP and WSDL. More information on REST can be found at the following website:

https://en.wikipedia.org/wiki/Representational_state_transfer

One example of a RESTful client is iAccess for Maconomy which 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 API exposed from Maconomy version 2.2 [2].

Maconomy also provides a Web Sockets access solution called OnePort. Web Sockets allows a client to open a full duplex communication socket with a webserver over a persistent TCP connection while conforming to firewall restrictions. More information on Web Sockets can be found at the following website:

https://en.wikipedia.org/wiki/WebSocket

Maconomy uses Web Sockets to allow the Workspace Client to communicate with the Coupling Service.

In this section, we will give a cursory overview of the technical architecture.

2.1 Technical Architecture

Figure 1 shows the high-level architecture of a Maconomy system providing access via a Reverse Proxy and the Coupling Service using iAccess as an example client. 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.

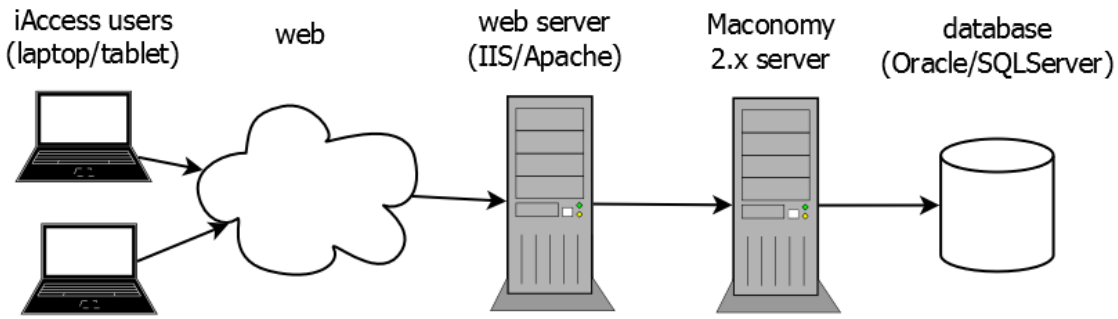

Figure 1: Architectural overview.

Maconomy 2.x server and database The Coupling Service requires a Maconomy 2.x server and database. OnePort is available from Maconomy 2.0 sp7 and 2.1.1. See the following section "OnePort (Workspace Client)" for more details on the web services. RESTful Web Services API for Maconomy is available from Maconomy 2.2 [2]. See the following section "RESTful Web Service API (iAccess & Integrations)" for more details on the web services.

Web server (IIS/Apache) One or more web servers are required to serve both static and dynamic content. 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 to the Coupling Service which exposes the RESTful Web Services API and OnePort Web Sockets. 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.

Clients Clients can be located both on the internal network or on the open Internet depending on the web server configuration and exposure.

iAccess Clients iAccess clients can run on different devices such as laptops with the main browsers (IE, Chrome, and Safari), as well as on iOS and Android tablets.

Workspace Client Workspace Client is a full featured Eclipse-based Java client for Maconomy that runs on both MacOS and Windows.

RestAPI Clients RestAPI Clients access Maconomy using the RESTful Web Services API. Examples are iAccess, TrafficLive and People Planer.

2.1.1 RESTful Web Service API (iAccess & Integrations)

As mentioned previously, Maconomy provides a 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 RestAPI endpoints and their usage:

/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 2.2.4 and 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 2.2.5 and 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.

Note that you only need to configure your Reverse Proxy to support RestAPI if you require these endpoints. You will need these endpoints if your installation requires iAccess, TrafficLive, People Planner, Touch RestAPI integration or a third party RestAPI integration.

2.1.2 OnePort (Workspace Client)

Maconomy provides a Web Socket solution for connecting a Workspace Client to a Coupling Service. For now, we will just briefly list the OnePort endpoints and their usage:

/handshake The *handshake* endpoint is part of the OnePort setup for Workspace Client. It is used to retrieve the basic information required by the Workspace Client to establish a connection with a Maconomy server instance. This information includes the company, database shortname, server version and allowed client versions.

/workspaces-rpc The *workspaces-rpc* endpoint is part of the OnePort setup for Workspace Client. It is used to provide a Web Sockets connection to the Coupling Service.

/analyzer The *analyzer* endpoint is part of the OnePort setup for Workspace Client. It is used to allow access to the Analyzer when using the OnePort setup for the Workspace Client.

Note that you only need to configure your Reverse Proxy to support OnePort if you require these endpoints. You will need these endpoints if your installation requires an HTTPS connection for the Workspace Client.

Chapter 3

Installing RESTful Web Service API

This section describes the installation process for the RESTful Web Service API. 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 Kona Space.

Note that you only need to install RestAPI and configure your Reverse Proxy to support RestAPI if your installation requires iAccess, TrafficLive, People Planner, Touch RestAPI integration or a third party RestAPI integration.

3.1 Security Considerations

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

3.1.1 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. For example, the iAccess login page will display a warning message in case HTTP is used instead of HTTPS to warn the user of this danger.

3.1.2 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 Configuration Procedures" for more details.

3.2 Prerequisites

The following are prerequisites to installing RESTful Web Services API:

- Any of the following Maconomy versions: 2.2 or higher
- The latest versions of MConfig. Currently version 8.18
- 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 a Maconomy application. For detailed instructions on setting up applications, see the Deltek Maconomy Installation Guide for your specific Maconomy version.

3.3 MConfig Installation

To begin installation with MConfig, complete the following steps:

Step 1 In the MConfig Main Window, double-click the application to open. You will be prompted to enter the database password for the instance. The Application Instance window displays. At the bottom will be a button to open "OSGi products" as shown in Figure 2.

Figure 2: The Application Instance window.

Step 2 Click OSGi products. The OSGi Server Selection screen appears as shown in Figure 3.

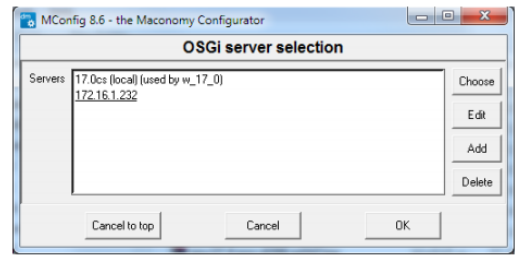

Figure 3: The OSGi Server Selection window.

Step 3 Select the Coupling Service to update and click "Choose".

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

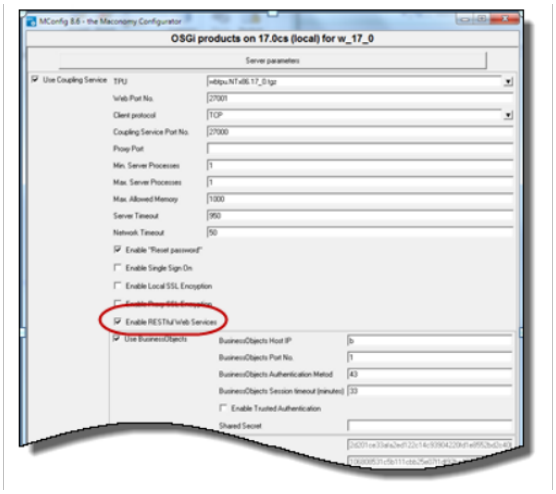

Figure 4: Enable RESTful Web Services.

Step 5 Click OK to save, and click OK at the SSL warning to return to the Application Instance window.

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.

Step 6 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.

Chapter 4

Installing OnePort

This section describes the configuration process for OnePort. 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 Kona Space.

Note that you only need to install OnPort and configure your Reverse Proxy to support OnePort if your installation requires an HTTPS connection for the Workspace Client.

4.1 Security Considerations

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

4.1.1 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.

4.1.2 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 Configuration Procedures” for more details.

4.2 Prerequisites

The following are prerequisites to configuring OnePort:

- Any of the following Maconomy versions: 2.0 sp7 or 2.1.1 or higher
- The latest versions of MConfig. Currently version 8.18
- If you are using IIS is must be version 8.0 or higher with WebSocket Protocol enabled in Server Roles
- If you are using Apache as the webserver it must be version 2.4 or higher, 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 a Maconomy application. For detailed instructions on setting up applications, see the Deltek Maconomy Installation Guide for your specific Maconomy version.

4.3 MConfig Installation

To begin installation with MConfig, complete the following steps:

Step 1 In the MConfig Main Window, double-click the application to open. You will be prompted to enter the database password for the instance. The Application Instance window displays. At the bottom will be a button to open “OSGi products” as shown in Figure 5.

Figure 5: The Application Instance window.

Step 2 Click OSGi products. The OSGi Server Selection screen appears as shown in Figure 6.

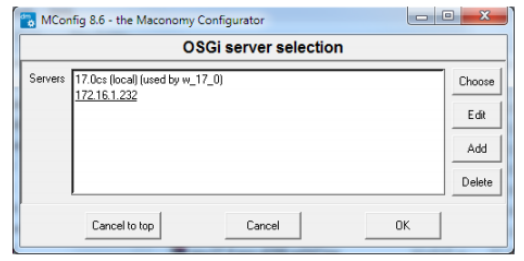

Figure 6: The OSGi Server Selection window.

Step 3 Select the Coupling Service to update and click "Choose".

Step 4 Choose "WebSockets" from the "Client protocol" drop down list, uncheck "Enable Local SSL Encryption" and check "Enable Proxy SSL Encryption" as shown in Figure 7.

Note: It is possible to setup the Reverse Proxy without SSL termination. You should not do this on a Production system because it will compromise the security of your system. However, for a test system, you might choose to leave SSL termination disabled on the Reverse Proxy. In this case, do not check "Enable Proxy SSL Encryption".

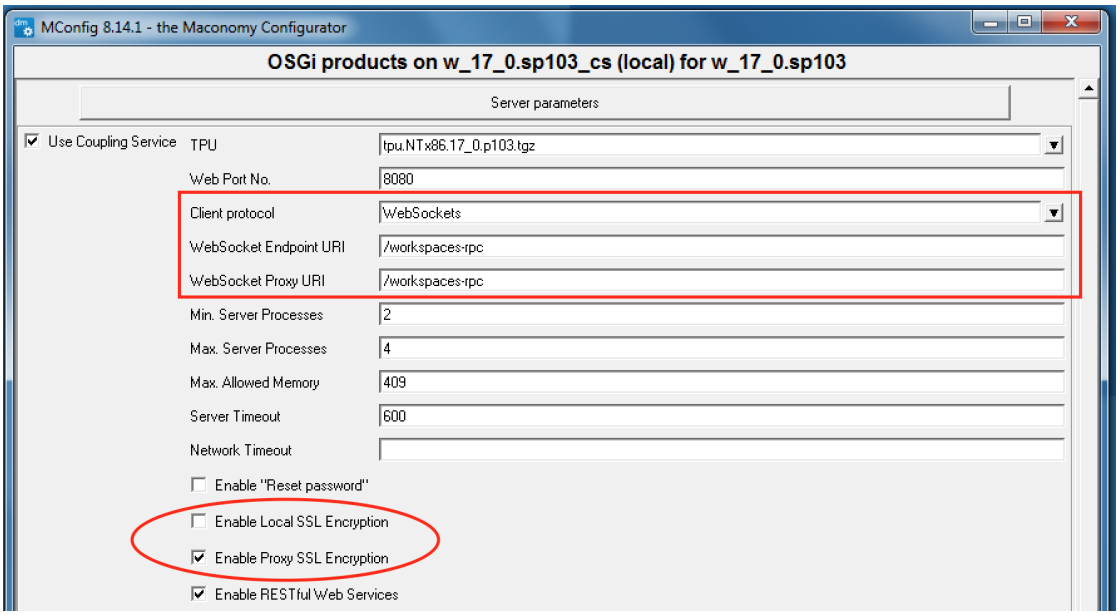

Figure 7: Turn on Web Sockets.

Step 5 Click OK to save, and click OK at the SSL warning to return to the Application Instance window.

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.

Step 6 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.

Chapter 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. If you do not already have a web server configured in MConfig, the entire setup of the IIS website will be accomplished automatically when you setup the Reverse Proxy using MConfig..

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

5.1 Enable IIS Support Automatically Using MConfig

Automatic IIS configuration requires MConfig 8.18 or higher. Previous version will not perform a correct configuration of IIS due to a shortcoming in MConfig. You can setup three different types of Reverse Proxies: iAccess, RestApi and OnePort. The automatic setup will create one IIS website for iAccess and a second website for the Web Server components of Maconomy (MScript, Portal and Web Services). See Figure 8. The iAccess Reverse Proxy rules will be applied to the iAccess website while the RestAPI and OnePort rules will be applied to the other website.

Note that all actions to setup the website will be completed including the setup for “cgi-bin”, “Application Pool”, “ISAPI and CGI Restriction” and “Handler Mappings”. If your Coupling Service is configured to “Enable Proxy SSL Encryption”, then the Reverse Proxy setup will include the required setup for HTTPS. You will need to install only an SSL certificate to complete the setup.

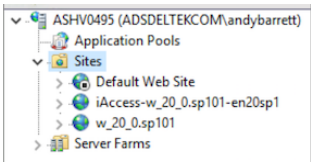

Figure 8: IIS Websites

5.1.1 Proxy Setup

1. Make sure WebSockets have been installed on your windows server if you intend to setup a OnePort Reverse Proxy. See "OnePort: WebSocket Setup IIS" for more details.
2. Install Microsoft Application Request Routing for IIS ARR. The easiest way to install ARR is to install and use Microsoft Web Platform Installer. Search for ARR and install the latest version.
3. Restart IIS Manager.

Note: You might find that you cannot install ARR using the Microsoft Web Platform Installer due to firewall restrictions. In that case, there is a single file download which contains all of the necessary components in one package: ARRv3_setup_amd64_en-us.EXE

5.1.2 MConfig Setup

To enable IIS support using MConfig, follow these steps:

Step 1 In the Application Instance window, click Web products as shown in Figure 9.

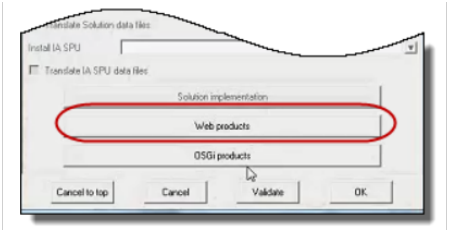

Figure 9: Select web products

Step 2 On the Web server selection screen, select the web server to update as shown in Figure 10, or add a new web server.

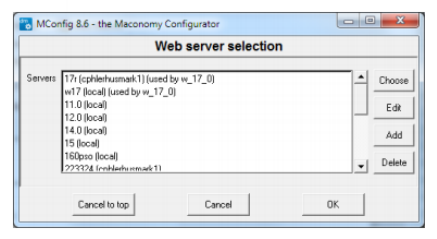

Figure 10: Select web server

Step 2a If you want to add a new web server, click Add. A new window will appear as shown in Figure 11. Give the web server a Descriptive name and select if the web server is local or not. Then click OK.

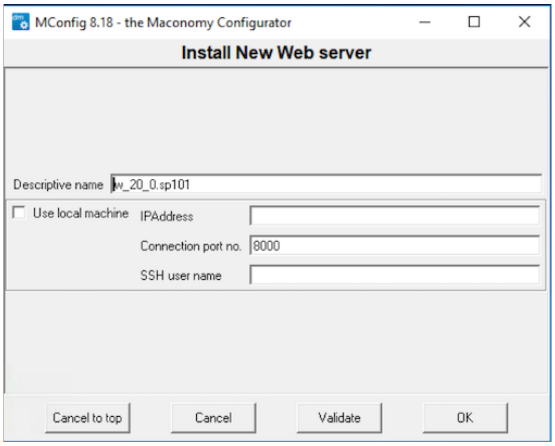

Figure 11: Add web server

Step 2b On the Web server selection screen, select the new web server and click Choose.

Step 2c Configure the Web Server Parameters as shown in Figure 12. Make sure that the physical web server home directory has been created on disk and click OK.

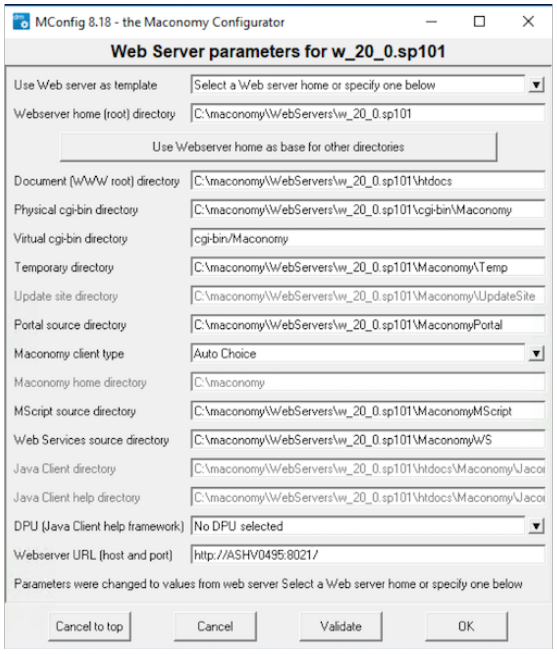

Figure 12: Web server parameters

Step 2d A new screen will appear. (See Figure 13) Select the check boxes to install whatever web products you require. If you want to install the Reverse Proxy for iAccess, you must install iAccess first.

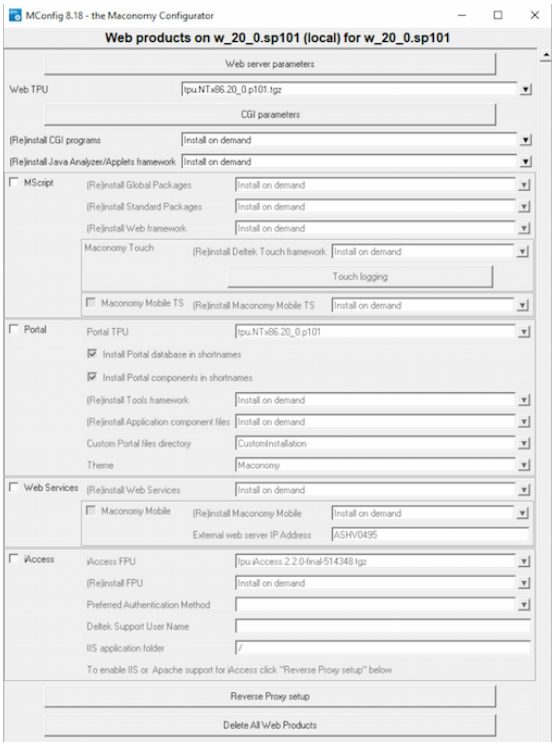

Figure 13: Web server components

Step 2e If you require the iAccess Reverse Proxy, select the iAccess check box as shown Figure 14. In the iAccess FPU field, select the relevant FPU from the drop-down list.

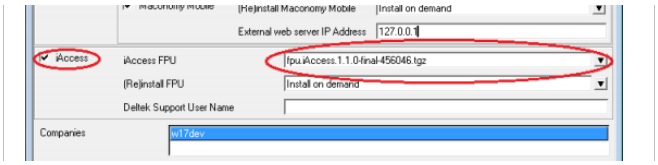

Figure 14: Web products window

Step 3 Click the Reverse Proxy setup button. A new screen will appear. (See Figure 15)

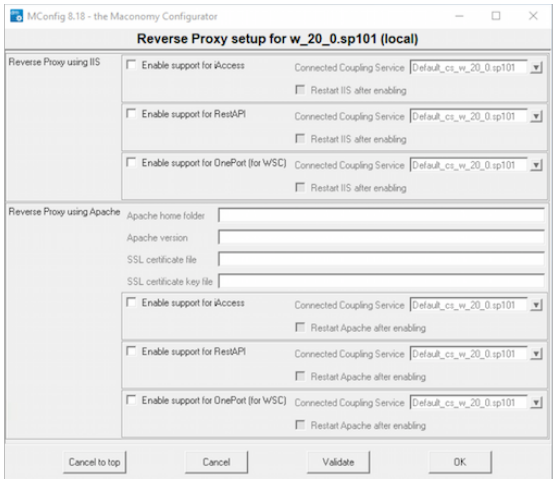

Figure 15: Reverse Proxy setup

Step 4 Select the check boxes for the IIS Reverse Proxies you require. You can also select the Coupling Service for the proxy and if you want IIS to be restarted automatically. Click OK.

Step 5 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.

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 for iAccess, MConfig also updates the IIS web.config file with a routing rule that ensures login pages and other non-root URLs load properly.

5.2 Enable IIS Support Manually Using IIS Manager

An alternative to installing iAccess is to configure IIS manually using IIS Manager.

5.2.1 Add the Site

Connect to your server in the Internet Information Services (IIS) Manager application and setup a web site. The site should have the files shown in Figure 16 as root files.

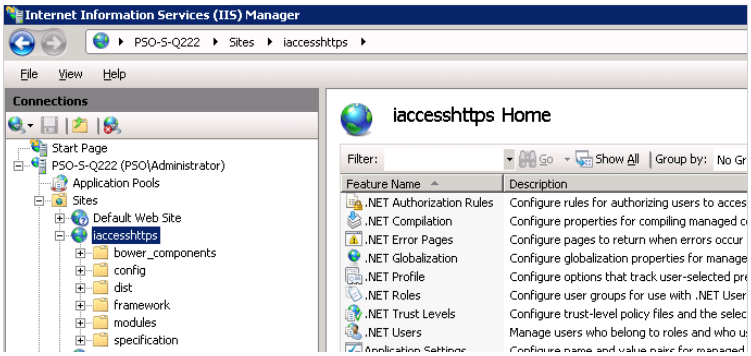

Figure 16: Add Site

5.2.2 Add MIME Types

This step is only required for iAccess.

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*.

5.2.3 Proxy Setup

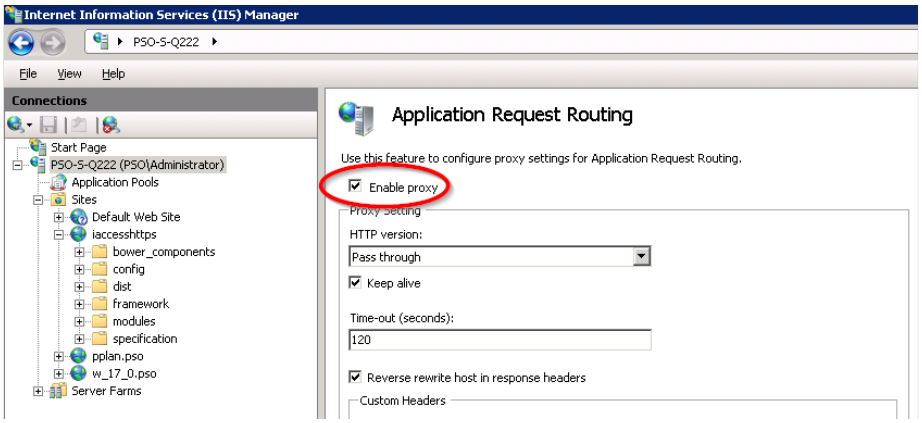

Figure 17: Enable Proxy

1. Make sure WebSockets have been installed on your windows server if you intend to setup a OnePort Reverse Proxy. See "OnePort: WebSocket Setup IIS" for more

details.

2. Install Microsoft Application Request Routing for IIS ARR. The easiest way to install ARR is to install and use Microsoft Web Platform Installer. Search for ARR and install the latest version.

3. Restart IIS Manager.

4. In the *Application Request Routing* configuration, click *server proxy settings*.

5. Check *Enable proxy* as shown in Figure 17.

6. 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 18.

Note: You might find that you cannot install ARR using the Microsoft Web Platform Installer due to firewall restrictions. In that case, there is a single file download which contains all of the necessary components in one package: ARRv3_setup_amd64_en-us.EXE

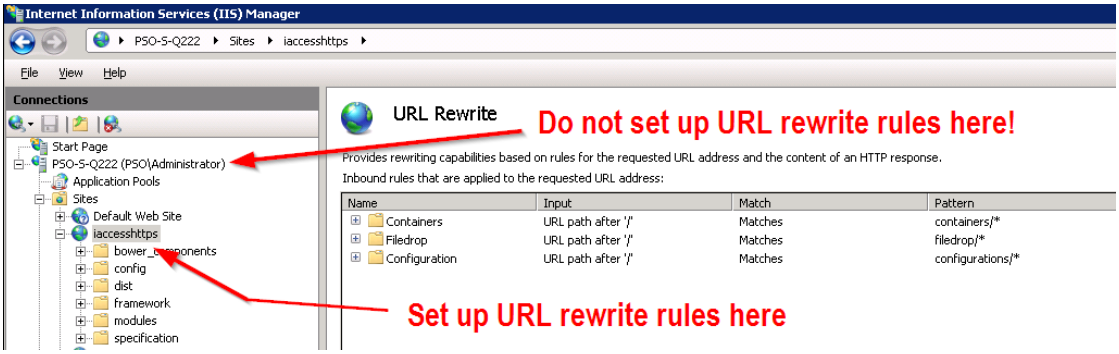

Figure 18: Add Proxy Rules

5.2.4 Edit Routing Rules

Deep Linking Support applies only to iAccess. However, if you choose to combine your Reverse Proxy rules into one web.config then you must add all endpoints into these rules.

To ensure that login pages (and other non-root URLs) load properly, open the IIS web.config 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>
```

5.2.5 Set up proxy for Containers API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in Figure 19.

Make sure to choose the Coupling Service webport when setting this up, for example, 4111 in Figure 19. 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 and Restful API requests from other clients like People Planner will fail.

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
```

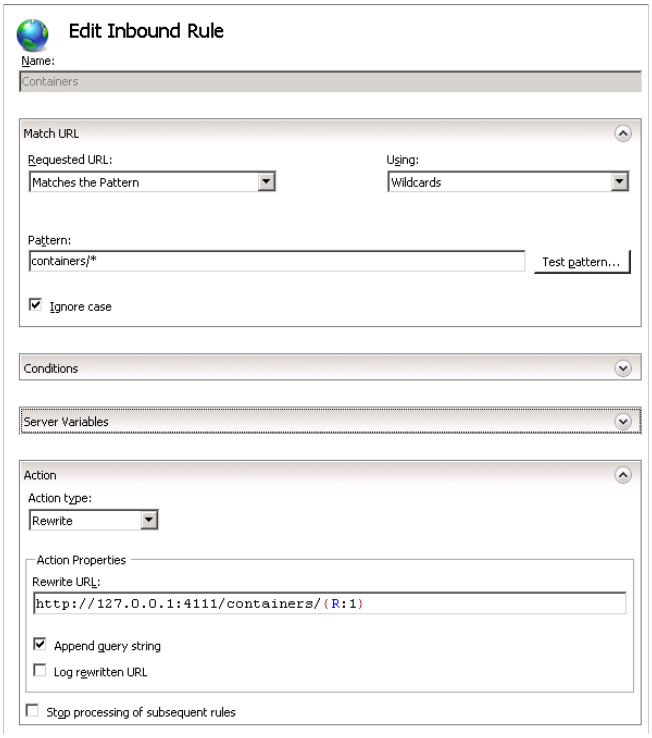

Figure 19: Containers API

5.2.6 Set up proxy for Configurations API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in Figure 20.

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
```

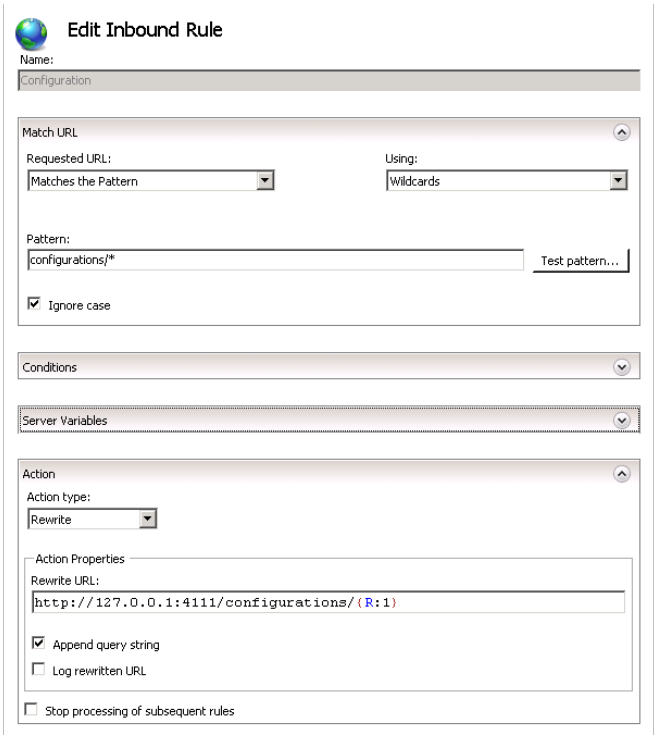

Figure 20: Configurations API

5.2.7 Set up proxy for Filedrop API

1. Click Add Rule...
2. Select Blank rule.
3. Fill out the rule as shown in Figure 21.

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
```

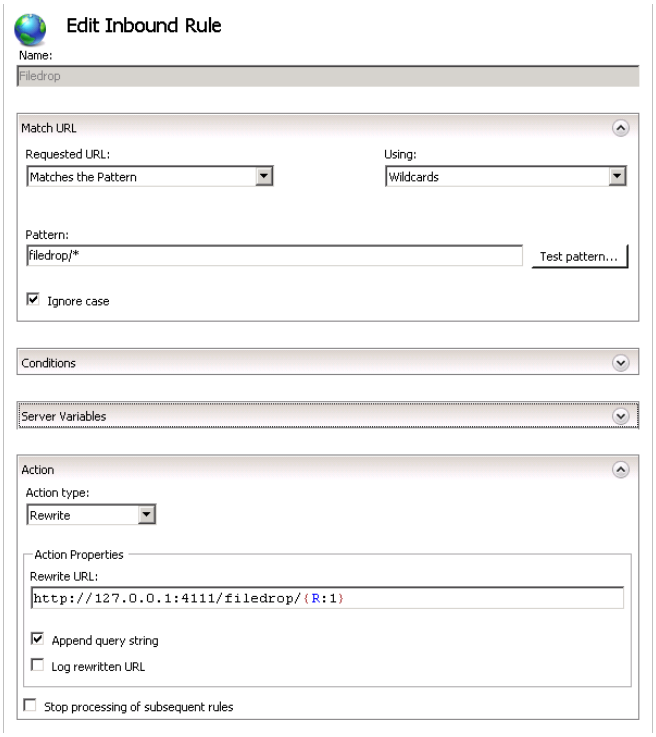

Figure 21: Filedrop API

5.2.8 Set up proxy for Auth API

1. Click Add Rule...
2. Select Blank rule.
3. Using Containers, Configurations or Filedrop as an example, fill out the rule for Auth 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
```

5.2.9 Set up proxy for Environment API

1. Click Add Rule...
2. Select Blank rule.
3. Using Containers, Configurations or Filedrop as an example, fill out the rule for Environment 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
```

5.2.10 Set up proxy for Handshake API

1. Click Add Rule...
2. Select Blank rule.
3. Using Containers, Configurations or Filedrop as an example, fill out the rule for Handshake with the following parameters:

Match URL

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

Action

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

5.2.11 Set up proxy for Web Sockets API

1. Click Add Rule...
2. Select Blank rule.
3. Using Containers, Configurations or Filedrop as an example, fill out the rule for Workspaces-RPC with the following parameters:

Match URL

```
Requested URL: Matches the Pattern
Using: Wilcards
Pattern: workspaces-rpc*
Ignore case: checked
```

Action

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

Note that IIS does not use `ws` or `wss` as the protocol for rewriting WebSockets in this configuration. The connection starts as an HTTP(S) connection with an "Upgrade: websocket" HTTP request header. The rewrite rule will create the same connection.

5.2.12 Set up proxy for Analyzer API

1. Click Add Rule...
2. Select Blank rule.
3. Using Containers, Configurations or Filedrop as an example, fill out the rule for Analyzer with the following parameters:

Match URL

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

Action

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

5.2.13 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.

Alternately, you could set `preserveHostHeader="true"` for `system.webServer/proxy` using the configuration editor in IIS Manager as shown in Figure 22.

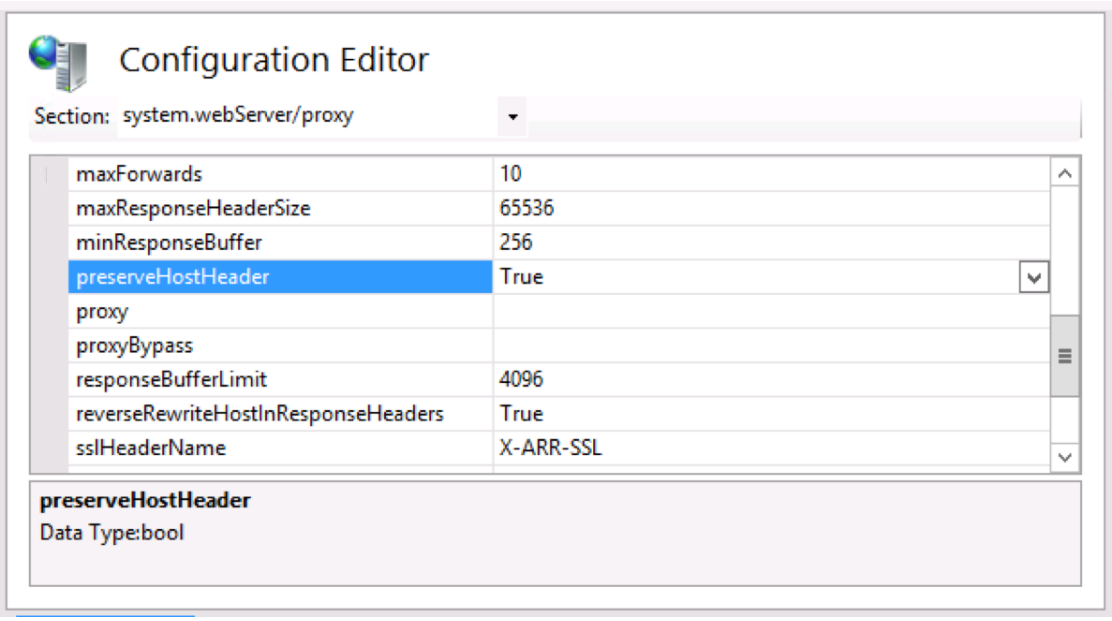

Figure 22: Configuration Editor preserveHostHeader

5.2.14 Set Up HTTPS

In IIS, Server variables can be used to access HTTP headers from the current request. Any HTTP header supplied by the current request is represented as a server variable that has a name generated in accordance to this naming convention:

- All dash (“-”) symbols in the HTTP header name are converted to underscore symbols (“_“).
- All letters in the HTTP header name are converted to capital case.
- “HTTP_” prefix is added to the header name.

For example, in order to access the HTTP header `X-Forwarded-Proto` from a rewrite rule, you can use the `HTTP_X_FORWARDED_PROTO` server variable. Set the server variable by doing the following:

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 23.

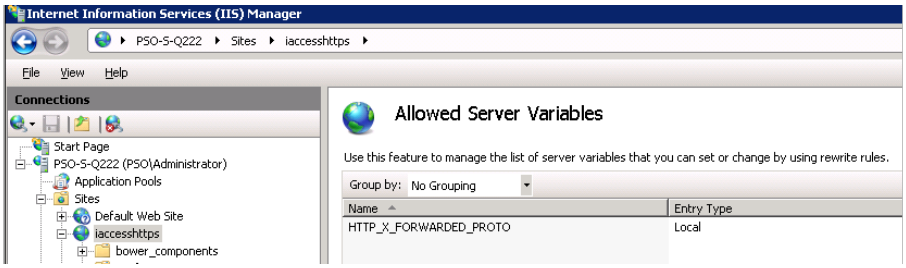

Figure 23: Add Server Variables

In the URL rewrite rules (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 24.

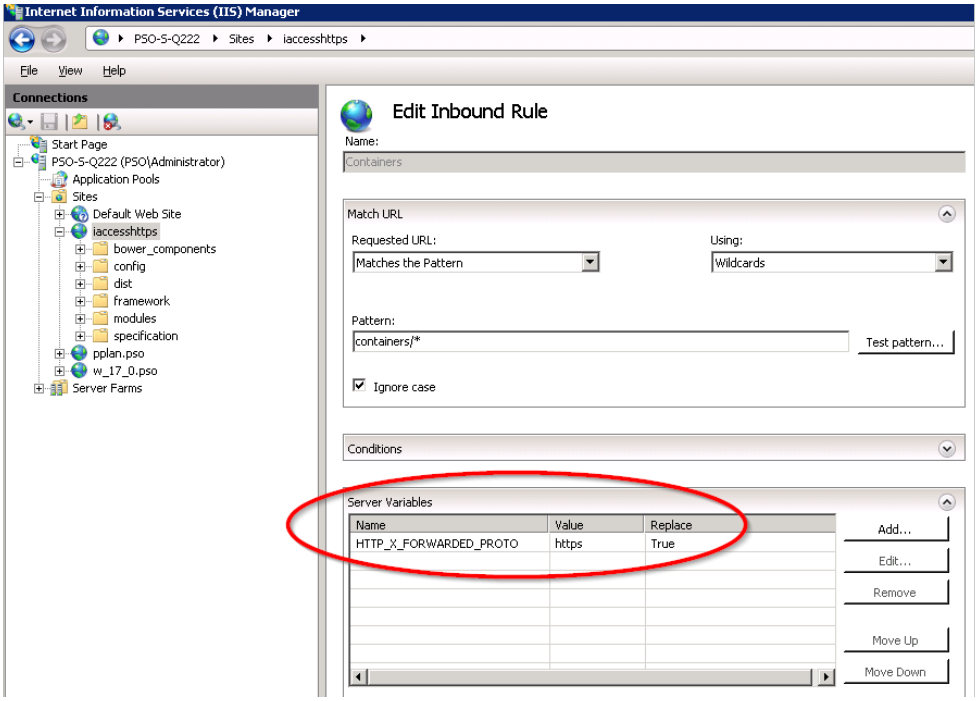

Figure 24: Setting up HTTPS

Restart the webserver.

Note: It is not possible to run both HTTP and HTTPS on the same IIS site using this configuration. See "Conditionally Setting Request Headers" under the "Additional Configuration Procedures" section for possible alternatives.

5.2.15 Example web.config

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <rewrite>
            <rules>
                <rule name="DeepLinkingSupport" stopProcessing="false">
                    <match url=".*" />
                    <action type="Rewrite" 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="environment ↩
*" negate="true" />
                    <add input="{REQUEST_FILENAME}" pattern="auth*"  ↩
negate="true" />
                    <add input="{REQUEST_FILENAME}" pattern="handshake*" ↩
 negate="true" />
                    <add input="{REQUEST_FILENAME}" pattern="workspaces- ↩
rpc*" negate="true" />
                    <add input="{REQUEST_FILENAME}" pattern="analyzer*"  ↩
negate="true" />
                </conditions>
            </rule>
            <rule name="Containers" patternSyntax="Wildcard">
                <match url="containers/*" />
                <action type="Rewrite" url="http://localhost:8081/ ↩
containers/{R:1}" />
                <serverVariables>
                    <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                </serverVariables>
            </rule>
            <rule name="Configurations" patternSyntax="Wildcard">
                <match url="configurations/*" />
                <action type="Rewrite" url="http://localhost:8081/ ↩
configurations/{R:1}" />
                <serverVariables>
                    <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                </serverVariables>
            </rule>
            <rule name="Filedrop" patternSyntax="Wildcard">
                <match url="filedrop/*" />
                <action type="Rewrite" url="http://localhost:8081/ ↩
filedrop/{R:1}" />
                <serverVariables>
                    <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                </serverVariables>
            </rule>
            <rule name="auth" patternSyntax="Wildcard">
                <match url="auth/*" />
                <action type="Rewrite" url="http://localhost:8081/auth/{ ↩
R:1}" />
                <serverVariables>
                    <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                </serverVariables>
```

```
                </rule>
                <rule name="environment" patternSyntax="Wildcard">
                    <match url="environment/*" />
                    <action type="Rewrite" url="http://localhost:8081/ ↩
    environment/{R:1}" />
                    <serverVariables>
                        <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                    </serverVariables>
                </rule>
                <rule name="handshake" patternSyntax="Wildcard">
                    <match url="handshake/*" />
                    <action type="Rewrite" url="http://localhost:8081/ ↩
    handshake/{R:1}" />
                    <serverVariables>
                        <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                    </serverVariables>
                </rule>
                <rule name="workspaces-rpc" patternSyntax="Wildcard">
                    <match url="workspaces-rpc*" />
                    <action type="Rewrite" url="http://localhost:8081/ ↩
    workspaces-rpc{R:1}" />
                    <serverVariables>
                        <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                    </serverVariables>
                </rule>
                <rule name="analyzer" patternSyntax="Wildcard">
                    <match url="analyzer/*" />
                    <action type="Rewrite" url="http://localhost:8081/ ↩
    analyzer/{R:1}" />
                    <serverVariables>
                        <set name="HTTP_X_FORWARDED_PROTO" value="https" />
                    </serverVariables>
                </rule>
            </rules>
        </rewrite>
        <staticContent>
            <remove fileExtension=".woff" />
            <mimeMap fileExtension=".woff" mimeType="application/font-woff" ↩
    />
        </staticContent>
    </system.webServer>
</configuration>
```

Chapter 6

Create a Website Using Apache

Here is a short guide to setting up iAccess and RestAPI Reverse Proxies on Apache (2.2 and 2.4). The guide also provides steps to setup a Reverse Proxy for OnePort on Apache 2.4. It is recommended to use the latest stable release.

Automatic Apache configuration requires MConfig 8.18 or higher. You can setup three different types of Reverse Proxies: iAccess, RestApi and OnePort. The automatic setup will create one Apache VirtualHost for iAccess and a second VirtualHost for the Web Server components of Maconomy (MScript, Portal and Web Services). The iAccess Reverse Proxy rules will be applied to the iAccess VirtualHost while the RestAPI and OnePort rules will be applied to the other VirtualHost.

Note that all actions to setup the VirtualHosts will be completed including the setup for the root directory and "cgi-bin". If your Coupling Service is configured to "Enable Proxy SSL Encryption", then the Reverse Proxy setup will include the required setup for HTTPS. You will need to install only an SSL certificate to complete the setup.

6.1 Download Apache

Download the Apache binary package including OpenSSL. Install it.

6.2 Enable Apache Support Using MConfig

Follow Steps 1 - 3 in "Enable IIS Support Automatically Using MConfig". You should be at the Reverse Proxy setup dialog.

Step 4 Enter your Apache install directory and version (2.2 or 2.4). Enter the path to your SSL Cert and Key if you are using SSL. Select the check boxes for the Apache

Reverse Proxies you require. See Figure 25. You can also select the Coupling Service for the proxy and if you want Apache to be restarted automatically. Click OK.

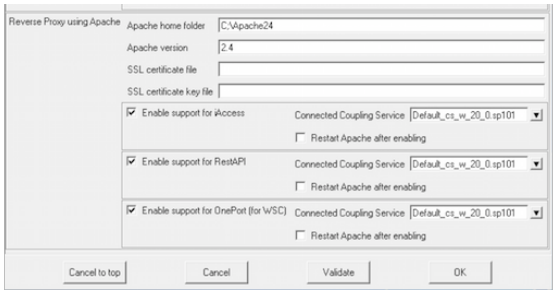

Figure 25: Reverse Proxy setup

Step 5 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.

Note: After you complete the initial installation with MConfig, you should check the setup in `httpd-vhosts.conf` and possibly modify parameters, such as the Listen port.

6.3 Enable Apache Support Manually

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 proxy_wstunnel_module modules/mod_proxy_wstunnel.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)

6.4 Enable Compression

In `httpd.conf`, do the following:

If using Apache 2.2, comment in the following module:

```
LoadModule deflate_module modules/mod_deflate.so
```

If using Apache 2.4, also comment in 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.

6.5 Setup with SSL

Configure your VirtualHosts in `conf/extra/httpd-vhosts.conf`.

Here is a template for setting up a virtual host that serves iAccess, Restful API and OnePort *with* SSL. Copy contents into the `httpd-vhosts.conf` file, and replace the variables with the desired values. Please observe that the line `Require all granted` below is only required for Apache 2.4.

For OnePort, note that wss:// is the secure websockets protocol corresponding to https:// while ws:// corresponds to http://

```
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>

    ScriptAlias /cgi-bin/ "<cgi-bin directory>"

    <Directory />
        Require all granted
        AllowOverride None
        Options None
    </Directory>

    <Proxy *>
```

```
        Order deny,allow
        Allow from all
        Require all granted
    </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

    ProxyPass  /handshake         http://<coupling-service-host>:<coupling- ↩
    service-web-port>/handshake        retry=0
    ProxyPass  /workspaces-rpc     ws://<coupling-service-host>:<coupling- ↩
    service-web-port>/workspaces-rpc      retry=0  disablereuse=On
    ProxyPass  /analyzer          http://<coupling-service-host>:<coupling- ↩
    service-web-port>/analyzer         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/w_20_0.101/htdocs/Maconomy/iAccess/ ↩
    w_20_0.101.w20/app"

    <Directory C:/Maconomy/Webservers/w_20_0.101/htdocs/Maconomy/iAccess/ ↩
    w_20_0.101.w20/app>
        Order deny,allow
```

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

    ScriptAlias /cgi-bin/ "C:/Maconomy/Webservers/w_20_0.101/cgi-bin/"

    <Directory />
        Require all granted
        AllowOverride None
        Options None
    </Directory>

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

    ProxyPass /handshake      http://localhost:8085/handshake      retry=0
    ProxyPass /workspaces-rpc  ws://localhost:8085/workspaces-rpc    retry=0 ↩
      disablereuse=On
    ProxyPass /analyzer       http://localhost:8085/analyzer       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>
```

6.6 Edit Routing Rules

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

Chapter 7

Verifying the Reverse Proxy Setup

To quickly verify the setup, open a browser (such as Chrome) and navigate to the following URL:

```
https://<external-url-and-port>/containers/v1/<shortname>
```

For example:

```
https://acme.com/containers/v1/w20sp100
```

If the proxy configuration is correct, the browser should either download, or show an XML or JSON document that looks like the following:

```
<Endpoint xmlns="http://www.deltek.com/ns/webservices" shortname="m20sp100" ↩
    authenticated="false">
  <Versions>
    <TPU major="20" minor="0" sp="100" fix="0" beta=""/>
    <APU major="20" minor="0" patch="100" hotfix=""/>
  </Versions>
  <Languages> ... </Languages>
  ...
  <Links>
    <Link href="https://acme.com/containers/v1/w20sp100" rel="self"/>
  </Links>
</Endpoint>
```

More extensive tests of the setup can be performed using tools like Postman or Paw:

```
https://www.getpostman.com
https://paw.cloud
```

Chapter 8

Additional Configuration Procedures

8.1 Conditionally Setting Request Headers

You might not want to always overwrite HTTP Request Headers, especially if the header might be sent by a client. For example, if you want to allow the client to specify whether the protocol is `http` or `https`, you can conditionally set the header if the client does not send `X-Forwarded-Proto`. The result is that the client and not the Reverse Proxy will control how URLs are constructed in the RestAPI response. If the client does not set `X-Forwarded-Proto` then it will be set by the Reverse Proxy to the default of `https`:

8.1.1 IIS

To conditionally set a Server Variable in IIS, uncheck `Replace the existing value` in the "Set Server Variable" dialog as shown in Figure 26.

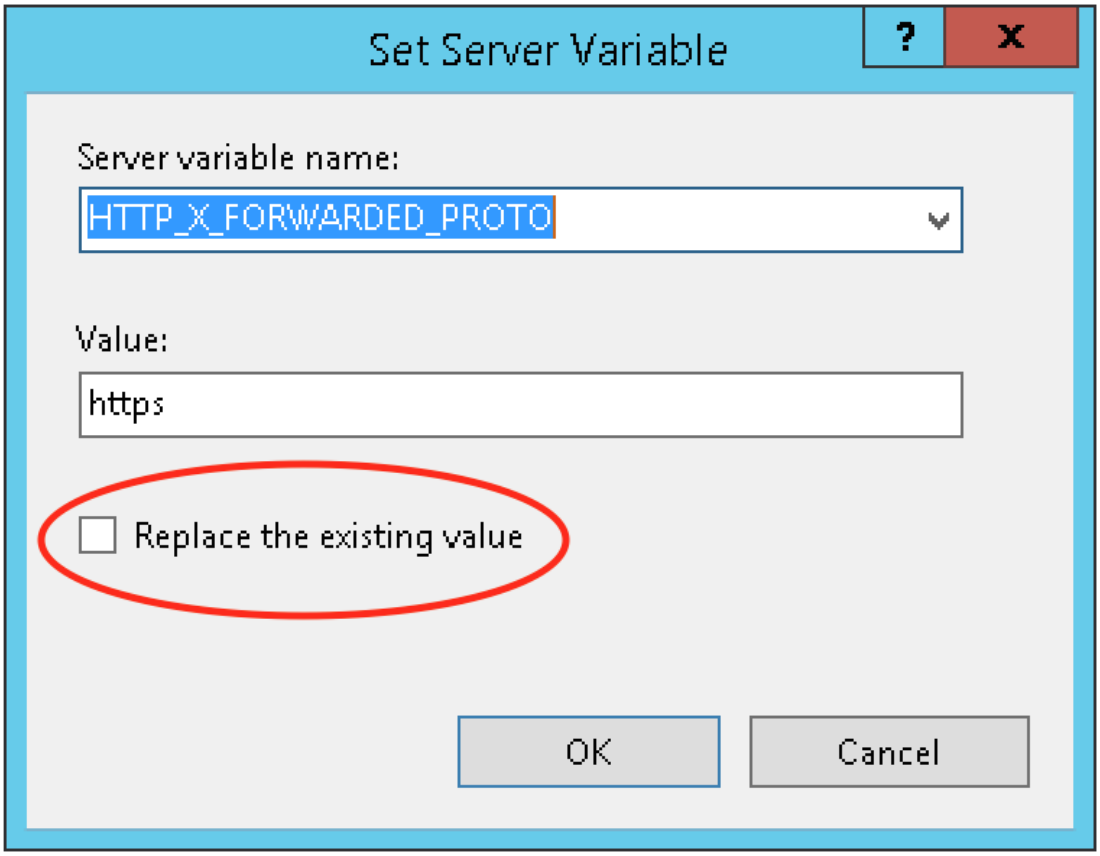

Figure 26: Conditionally Set Server Variable

8.1.2 Apache

Add the following lines to your server configuration file.

In Apache prior to 2.4.7:

```
SetEnvIf X-Forwarded-Proto "^$" no_x_forwarded_proto
RequestHeader set X-Forwarded-Proto "https" env=no_x_forwarded_proto
```

In In Apache 2.4.7 and higher:

```
RequestHeader setIfEmpty X-Forwarded-Proto "https"
```

8.2 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: default-src 'self', frame-ancestors 'self'
X-Frame-Options: SAMEORIGIN
```

These headers are then added to all responses.

8.2.1 IIS

Set the response headers for the `system.webServer/httpProtocol` using the configuration editor in IIS Manager as shown in Figure 27.

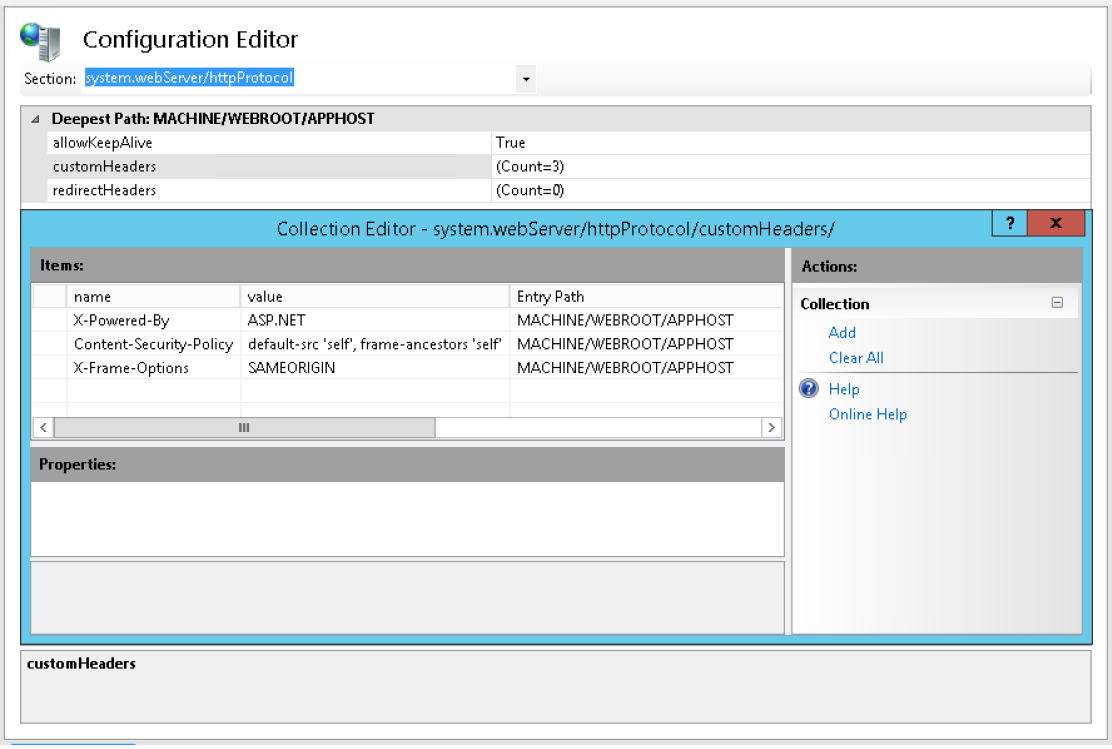

Figure 27: Configuration Editor Response Headers

8.2.2 Apache

Add the following lines to your server configuration file:

```
Header set Content-Security-Policy "default-src 'self', frame-ancestors ↩
'self'"
Header set X-Frame-Options "SAMEORIGIN"
```

8.3 Domain Login and Single Sign On

If your Maconomy instance is configured to use Domain Login through Single Sign On, you might get an error when trying to access Maconomy using the RestAPI. Your RestAPI client application should send an HTTP Request Header to inform the Coupling Service to use Maconomy logon credentials instead of domain credentials. You will also have to pass the Maconomy login and password using Basic Authentication. Set the following:

```
Maconomy-Authentication: X-Basic, X-Reconnect, X-Force-Maconomy-Credentials
```

Although each RestAPI client should send this header, some do not. The Deltek products People Planner and Traffic Live do not send this header and there is currently no way for this header to be configured in the RestAPI requests these products use to integrate with Maconomy. The easiest way to overcome this limitation is to have the Reverse Proxy set this HTTP Request Header.

Note: People Planner 3.6.1 and higher and 3.7 and higher have been enhanced to send the X-Force-Maconomy-Credentials header properly.

8.3.1 IIS

Set the server variable `HTTP_MACONOMY_AUTHENTICATION` to `X-Force-Maconomy-Credentials` as shown in figure 28.

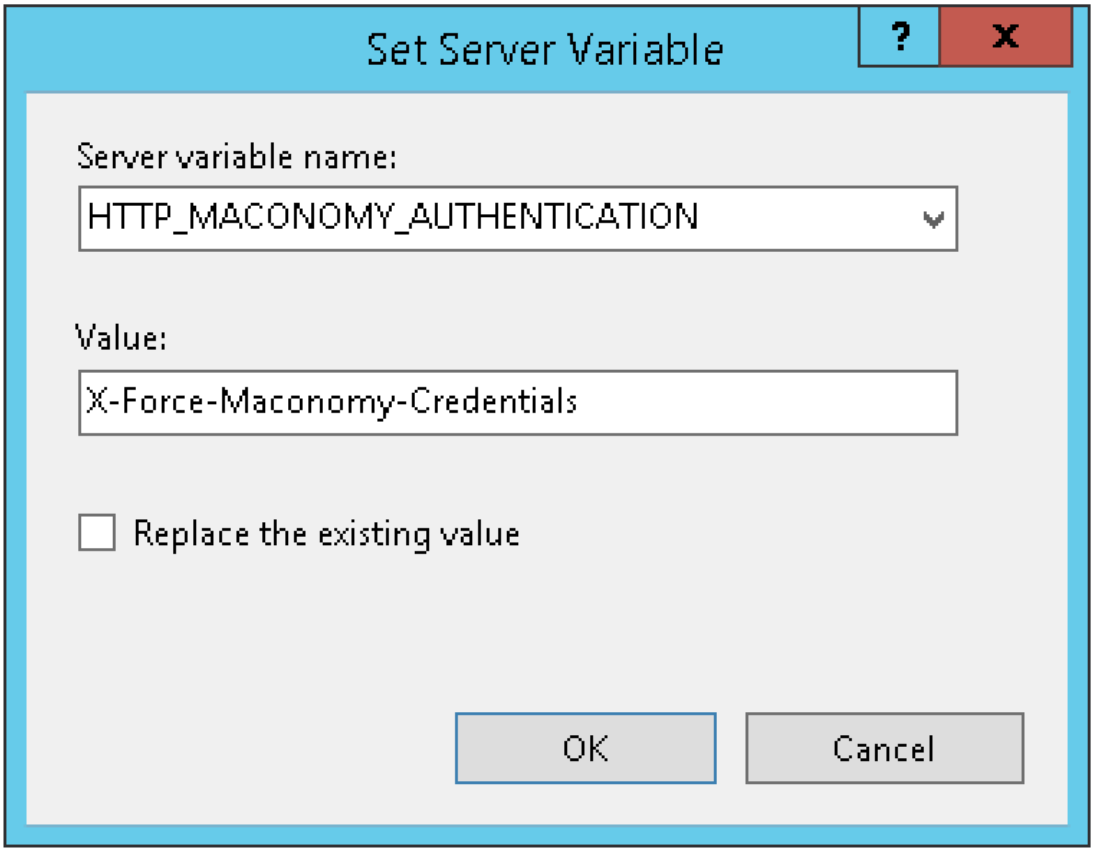

Figure 28: Set HTTP_MACONOMY_AUTHENTICATION

8.3.2 Apache

Add the following line to your server configuration file:

```
RequestHeader append Maconomy-Authentication "X-Force-Maconomy-Credentials"
```

8.4 OnePort: WebSocket Setup IIS

WebSockets are not enabled by default for Windows servers or in IIS. In order to use IIS to proxy WebSocket requests, you must first enable Websocket support in the OS and also in IIS. It should also be noted that WebSockets were first supported in IIS version 8.0. See the following Microsoft documentation:

```
https://docs.microsoft.com/en-us/iis/configuration/system.webserver/ ↩
    websocket
```

8.5 OnePort: Timeout and Keep-Alive Settings for IIS

When using IIS as the Reverse Proxy for WebSockets, two settings need to be changed so that the WebSocket connection is not terminated by a timeout. Use the configuration editor in IIS Manager to set the `system.applicationHost/webLimits` for `connectionTimeout` equal to `00:05:00` and `minBytesPerSecond` equal to `0` as shown in Figures 29 and 30 respectively.

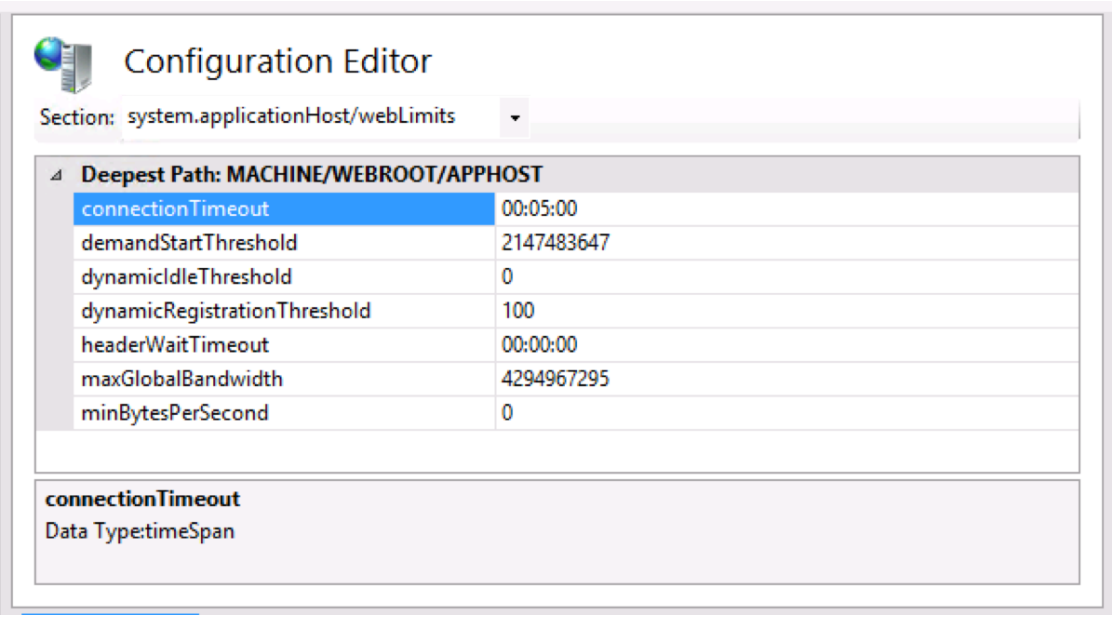

Figure 29: Configuration Editor connectionTimeout

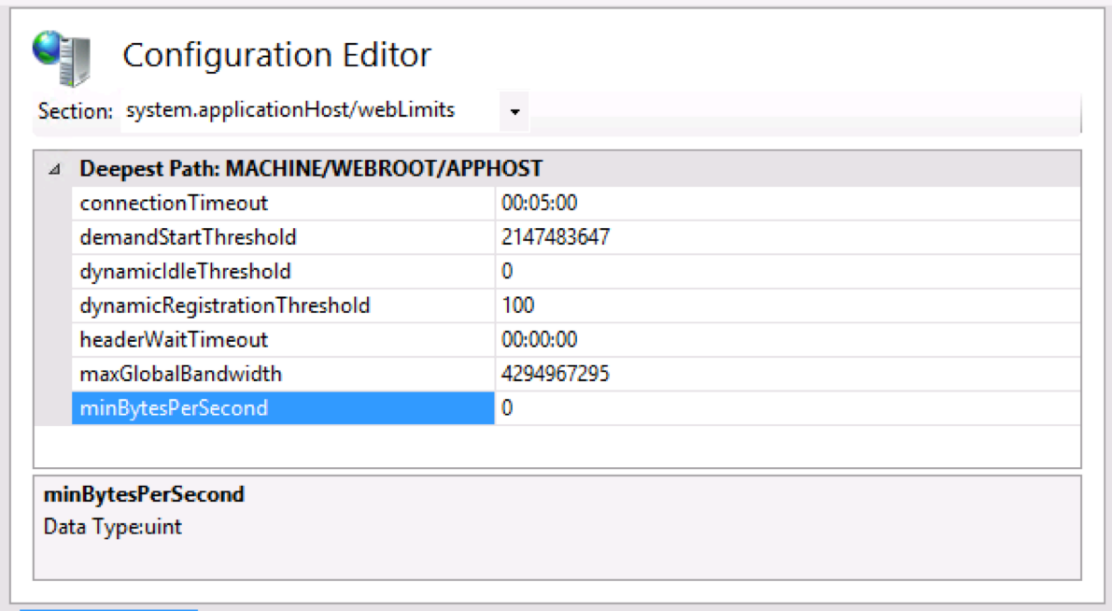

Figure 30: Configuration Editor minBytesPerSecond

The dirmi keep-alive needs to be set in the Coupling Service configuration file `server.ini` to 4 minutes so that the keep-alive fires before the 5 minute IIS connectionTimeout:

```
coupling.dirmi.ping-interval-millis = 240000
```

8.6 OnePort: Set Web Socket Endpoint

The Coupling Service configuration file `server.ini` can be modified to set the URI where the Maconomy Coupling Service is exposed when using websockets transport. The path component of this URI will be used as the Servlet alias for the web sockets servlet.

By default, the value of this configuration parameter is sent to clients by the handshake web service. To override the default URI sent in the handshake, use the option coupling.dirmi.websockets-uri.

Here is an example of changing the URI:

```
coupling.dirmi.websockets-uri=/workspaces-rpc-test
```

The resulting handshake would be the following:

```
"dirmi": {
    "websockets": {
    "endpoint": "/workspaces-rpc-test",
    "maxUnconsumedBytes": 8388608
    }
```

8.7 OnePort: Set Client Handshake Web Socket URI

The Coupling Service configuration file `server.ini` can be modified to set the websockets URI which will be sent in the handshake web service. By default, the value of this configuration parameter is derived from the value of the option: coupling.dirmi.websockets-uri

Use this option to override the URI sent in the handshake. This can be useful if, for example, an intermediate proxy is set up to perform URI rewriting. To override the default URI sent in the handshake, use the option coupling.dirmi.websockets-uri.

Here is an example of changing the handshake URI:

```
coupling.dirmi.client.websockets-uri=/workspaces-rpc-new-endpoint
```

The resulting handshake would be the following:

```
"dirmi": {
    "websockets": {
    "endpoint": "/workspaces-rpc-new-endpoint",
    "maxUnconsumedBytes": 8388608
    }
```

8.8 OnePort: Force Protocol, Host and Port

Changing these settings is unnecessary when using either IIS or Apache. But, these setting could be useful if you choose a different Reverse Proxy system.

The Coupling Service configuration file `server.ini` can be modified to force all web service redirects and hyperlinks to use a particular protocol (URI scheme), host and port.

These options should only be used if the reverse proxy can not be configured in the following way:

1. The proxy must preserve the “Host” HTTP request header
2. The proxy must add the “X-Forwarded-Proto” HTTP request header to indicate the protocol of the originating request (e.g. https)

These options can be used to force redirects and hyperlinks to be generated using a particular protocol, host, and port:

```
web.force-protocol = <protocol>
web.force-host = <host>
web.force-port = <port>
```

Note that 'web.force-port' will only take effect if 'web.force-host' is specified.

Here is an example of setting these options:

```
web.force-protocol = https
web.force-host = 127.0.0.1
web.force-port = 8888
```

A resulting RestAPI URI would look like the following:

```
"links": {
"self": {
  "href": "https://127.0.0.1:8888/containers/v1/en17sp3d/employees/data; ↩
    employeenumber=11",
  "rel": "self"
}
```

8.9 OnePort: Required for Maconomy 2.3.3 and 2.4.0

The Workspace Client and Coupling Service in Maconomy version 2.3.3 and 2.4.0 do not support TCP DIRMI connections anymore: only DIRMI WebSocket (OnePort) style connections are supported. This means that in MConfig it is not possible to select TCP DIRMI connections for 2.3.3 and 2.4.0 or higher Maconomy versions. Setups using TCP DIRMI must be reconfigured to use DIRMI WebSockets (OnePort) after upgrading to 2.3.3 or 2.4.0.

The Workspace Client and the Java Analyzer in Maconomy version 2.3.3 and 2.4.0 *do* check HTTPS certificates involved in HTTPS communication from the Workspace Client and the Java Analyzer. This means that the old mechanism for using "Local SSL encryption" is no longer supported. Any SSL termination is expected to be done by a Reverse Proxy in front of the Coupling Service as described in this manual.

To make it easier to spot HTTPS certificate related errors, the Workspace Client marks errors related to HTTPS certificate checking by including the "certificate related" text in the error message shown to the user.

See the release notes for these Maconomy versions for more information.

8.9.1 MConfig Details

Because only WebSocket DIRMI "mode" is supported, in MConfig it is not possible to select TCP DIRMI and SSL termination must be done using a Reverse Proxy.

The following settings that have been removed:

1. Under “OSGi Products”, the “Client protocol” popup has been removed as web-sockets is now the only supported variation of DIRMI
2. Under “OSGi Products”, the “Enable Local SSL encryption” checkbox has been removed as all encryption is must be done using a Reverse Proxy.
3. In the Coupling Service server.ini file, the “server.public-keystore” and “server.private-keystore” settings have both been removed as they are not needed anymore.
4. In the Coupling Service server.ini file, the “coupling.dirmi.encryption” settings have been removed as it’s not needed anymore.

8.9.2 SSL/TLS Details

1. The Workspace Client supports only TLS version 1.2 in 2.3.3 and 2.4.0.
2. The Workspace Client supports the following TSL cipher suites in 2.3.3 and 2.4.0:
   - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
   - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
   - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
   - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
3. The SSL / TLS compression option(s), as a general rule, should be disabled.
4. Please take special care to have procedures in place to renew the certificate(s) used, otherwise access to the system will be denied when the certificate validity period expires. Also, *do* have procedures in place to revoke the certificate(s) in use if the Reverse Proxy server is breached and the certificate private key is stolen.

Host name / IP address checking

The Workspace Client in 2.3.3 and 2.4.0 now checks the hostname and / or IP-address of the server address / hostname used against the certificate as part of the HTTPS certificate checks. This means that a certificate appropriate for the server address / hostname must be used.

Validity checking

The Workspace Client in 2.3.3 and 2.4.0 now checks validity of the server certificate as part of the HTTPS certificate checks. This means that a valid server certificate must be used at all times.

OCSP / CRL checking

The Workspace Client in 2.3.3 and 2.4.0 now checks whether the server certificate has been revoked as part of the HTTPS certificate checks. This check is only done if the certificate contains OCSP and / or CRL information. This check can be disabled by editing the Maconomy.ini file of the Workspace Client and removing the following options:

```
-Dcom.sun.security.enableCRLDP=true
-Dcom.sun.net.ssl.checkRevocation=true
```

The changed Workspace Client must then be distributed to the client machines. Disabling the OCSP / CRL checks can be useful if the network performance impact of the OCSP / CRL checks is too high.

Distributing self-signed certificates to client machines

The Workspace Client uses the list of certificates from the personal OS certificate stores on Mac OS X / macOS / Windows to determine trust. This implies that if the certificate used by a Maconomy system is a self-signed certificate which cannot be validated using the existing (root) certificates in the certificate stores of Mac OS X / macOS or Windows, then that certificate or it's root issuer needs to be added to those.

Chapter 9

Troubleshooting Guide

The following section provides clues to solve common issues. If you cannot find a solution to your specific problem, please post a conversation in the *Maconomy RESTful Web Services* or *iAccess for Maconomy* Kona spaces 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 31 shows Developer Tools in Chrome where the Network Tab can be a very powerful tool to uncover installation and network problems.

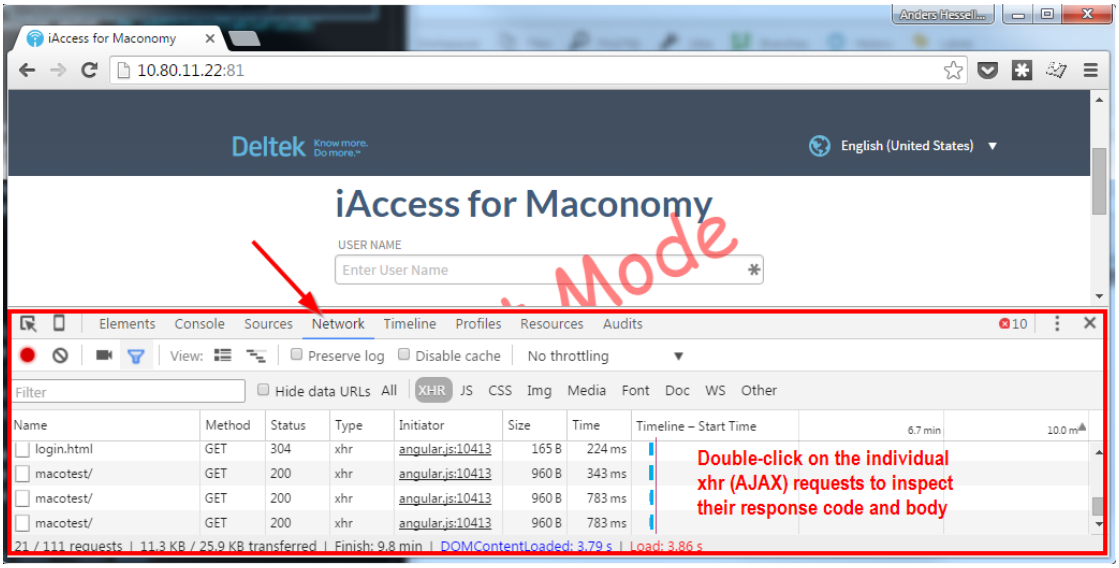

Figure 31: Use the Network Tab in your browser's Developer tools to debug failing requests and login problems.

"*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 "Create a Website Using IIS" 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 "Create a Website Using IIS" 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.

Error 500 in the Browser on Apache 2.2 Installations

If you get a 500 error code in the browser and the following message in the Apache error log: `configuration error: couldn't perform authentication. AuthType not set!: /`, the cause is an incorrect `vhosts.conf` file. Specifically, make sure that the line `Require all granted` is not present. This problem occurs for Apache 2.2 only. Apache 2.4 requires this line.

"*Domain login failed*"-Error

When accessing the RESTful Web Service API, if SSO is enabled for the Maconomy instance, a domain login failed error message could be returned to the client:

```
{
"errorMessage": "Domain login failed",
"errorFamily": "service",
"errorSeverity": "error"
}
```

The HTTP Header "Maconomy-Authentication" must include the option "X-Force-Maconomy-Credentials" so that basic authentication will be used instead of SSO for RestAPI requests:

```
Maconomy-Authentication: X-Basic,X-Reconnect,X-Force-Maconomy- ↩
Credentials
```

See "Domain Login and Single Sign On" under the "Additional Configuration Procedures" section for options on how to configure this header.

"*Connection could not be established: Bad Sec-WebSocket-Accept (1002)*"-Error

If you are using a OnePort Reverse Proxy, this error will manifest itself by the WSC starting and showing the Server Configuration dialog. When the connection is attempted, the WSC just closes. In the log file for the WSC, you will see the above error message. This error message means that WebSockets have not been enabled for the server. See "OnePort: WebSocket Setup IIS" for more details.

You might also see this error if you have not configured your firewall properly to allow bidirectional communication over the socket connection. You must establish both inbound and outbound rules in your firewall to allow WebSockets to function properly. In addition, the WebSocket connection returns HTTP response headers that also must be allowed through the firewall.

Chapter 10

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.

10.1 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.

10.2 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.

10.3 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%2FPubData%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 11

Figures

(1) Architectural Overview
(2) The Application Instance window
(3) The OSGi Server Selection window
(4) Enable RESTful Web Services
(5) The Application Instance window
(6) The OSGi Server Selection window
(7) Turn on Web Sockets
(8) IIS Websites
(9) Select web products
(10) Select web server
(11) Add web server
(12) Web server parameters
(13) Web server components
(14) Web products window
(15) Reverse Proxy setup
(16) Add Site
(17) Enable Proxy
(18) Add Proxy Rules
(19) Containers API

(20) Configurations API

(21) Filedrop API

(22) Configuration Editor preserveHostHeader

(23) Add Server Variables

(24) Setting up HTTPS

(25) Reverse Proxy setup

(26) Conditionally Set Server Variable

(27) Configuration Editor Response Headers

(28) Set HTTP_MACONOMY_AUTHENTICATION

(29) Configuration Editor connectionTimeout

(30) Configuration Editor minBytesPerSecond

(31) 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.