MVSC - Multivia Sm@rt-Connect

User manual

ATRUVIA

Genossenschaftliche FinanzGruppe
Volksbanken Raiffeisenbanken

MVSC - Multivia Sm@rtConnect

User manual

Version: 4.5
Datum/Uhrzeit: 28.10.2025 / 13:28 Uhr

Revision History

Nummer	Datum	Inhalt / Änderungen
12	21.10.2025	Addition for MVSC release 4.5: • The chapter "Use" has been renamed to "Using MVSC". In addition, it has been expanded to include the subchapter "Verification of payee".
11	17.10.2024	Addition for MVSC release 4.0: • The "introduction", chapter "General usability of MVSC", "Short description of MVSC" has been extended by a description of how to use Java version 1.8 or 17. Two web addresses were also included for downloading the desired MVSC version. • The Chapter "Use in console mode", section "Call from the console" has been adjusted. The description of the call variant "D" has been divided into two sections: The description of the order types for downloads and the description of the order types for uploads. The description of the download order types has been added.
10	28.08.2023	Addition for MVSC release 3.5.0: • The section "Authentication" of the chapter "Setup", "Application start" has been extended. After the first login, the initial password has to be changed.
09	16.11.2022	Addition for MVSC release 3.0.0: • The chapter "Technical logging" has been adjusted: the figure of the menu "Help", "Logging" has been renewed and can now be found in the section "Log level". The section "Menu item" has been deleted. • Chapter "Help": The reference is now made to the manual. The description of the "help texts" has been deleted.

Nummer	Datum	Inhalt / Änderungen
08	13.05.2022	Addition for MVSC release 3.0.0: • The chapter "Structure of the program" has been extended by the new tab "Order History". In addition, the "Configuration" tab has been renamed to "Accesses". This avoids confusion with the menu item of the same name. • Chapter "Specifying EBICS access data", section "Setup process": the description of the recommended signature medium "certificate" has been added. In addition, notes were included: from the EBICS version "EBICS 3.0" no new chip cards and security files can be generated or initialized. • Chapter "Specifying EBICS access data" has been extended by the new section "Change key". • Chapter "Check options" has been expanded to include the section "Order History". • Chapter "Supported signature procedures", section "Security file": the warning about entering the wrong password has been corrected. • Chapters "Use in console mode", "Automated use via batch file" and "Container creation": the call variants of MVSC has been revised.
07	11.03.2021	Addition for MVSC release 2.6.0: • A brief description of the difference between "MVSC Vollversion" and "MVSC Sign" has been included in chapter "General usability of MVSC" of the "introduction". • A reference to "MVSC Sign" has been added to chapter "Structure of the program".
06	04.03.2020	Addition for MVSC release 2.6.0: • Chapter "Data transfer via the GUI", section "Executing an upload order type ": a tip for using the order type "AUTO" has been added • Chapter "Use in console mode" has been expanded to include the section "Order types 'AUTO' and 'AUTD'
05	19/09/2019	Addition for MVSC release 2.5.0: • The introduction has been extended by the new section "Multilingualism". • Chapter "Supported signature procedures", section "Security file": warning on five consecutive incorrect password entries added • The chapter "Directory structure" has been extended by the directory "Original". • Into the chapter "Installation" the section "Update of the signature version of an access ID" has been added. In addition, the selection of the default language was added in the section "Installation with the wizard 'install.jar'". • Chapter "Specifying EBICS access data", section "Setup process": the description of the setup of an EBICS access has been added with the specification of the length and the permissible characters. In addition, some screenshots have been exchanged because the order of the signature versions has changed. • Chapter "Default settings", section "Group 'Other settings'": new option "Send selected sepa-files as IBAN-Only" added
04	11/10/2018	Addition for MVSC release 2.0.3: • Chapter "General applications of MVSC" and chapter "System requirements": a Java version 1.7 or 1.8 is required

Nummer	Datum	Inhalt / Änderungen
03	16/08/2018	Addition for MVSC release 2.0.2: • Screenshots updated due to a new logo • Chapter "Licence server" newly added • Return values in console mode: return value 25 added • Chapter "Use in console mode", section "Call from the console": call parameters of the call variants "C" and "D" adjusted
02	03/11/2017	MVSC 2.0 manual based on MVSC 1.0
01	03/08/2013	MVSC 1.0 manual based on PCTI 2.0

Öffentlich (C1)

Table of Contents

List of Figures

List of Tables

Introduction

1. General usability of MVSC

Short description of MVSC

MVSC is a Java-based tool for secure transfers of already created order files via the EBICS procedure. The application can be executed in the two modes "Data transfer via the GUI" and "Console call", which enable a needs-oriented definition of the software.

In contrast to the 'MVSC full version', 'MVSC Sign' can only be used to sign order files. Sending and collecting order files is not possible with 'MVSC-Sign'. The tab 'data transfers' is only available in the full version of 'MVSC' and not in 'MVSC Sign'.

MVSC benefits from the platform independence given in Java and can be executed on all operating systems on which a Java version 1.8 or 17 is installed.

There are 2 versions of MVSC, one for Java 1.8 and one for Java 17, so there are also two web addresses for downloading the version of MVSC.

If you want to use the MVSC version for **Java 1.8**, go to the following web address:

"https://smartconnect.multivia-suite.de/software/Install_MVSC.jar".

If you want to use the MVSC version for Java 17, go to the following web address:

https://smartconnect.multivia-suite.de/software/Install_MVSC17.jar

Note: Updates are installed automatically if the respective MVSC version is already available.

Multilingualism

In the upper right area are buttons for selecting the language.

This is shown in the following figure:

Figure 1. Language selection in the upper right area

2. EBICS basics

Definition

EBICS (Electronic Banking Internet Communication Standard) is the name of a multi-bank capable standard for transferring payment files via the Internet protocols TCP/IP, HTTP and HTTPS. EBICS is considered the successor of the previously existing FTAM standard "RDT with customers", which communicated with the bank server through direct dial-in via ISDN and/or DATEX-P.

Security aspects

Many FTAM characteristics were retained, such as the data model (customer / user / account) and the activation procedure (INI letter). The electronic signature from FTAM is also supported by EBICS. This makes a migration to the new EBICS procedure possible for former FTAM customers. Aside from the electronic signature supported by FTAM so far, which is stored in the form of a security file, EBICS offers the additional option to store the electronic signature on a smartcard. More information on the setup of an EBICS access under MVSC can be found in the chapter "Setup".

Legal framework

Since 1 January 2008, financial institutions are obligated to support the EBICS standard, while the same obligation regarding the FTAM procedure ended on 31/10/2010. (http://www.ebics-zka.de/).

Chapter 1. Functions

1.1. General information on functionality

Call variants

The GUI of MVSC offers all required functions, from the creation of EBICS and Internet connection data all the way to data transfers of all kinds.

After the EBICS access has been activated on the GUI, the use of the console mode is only a matter of calling the application.

1.2. Supported signature procedures

Signature procedures

EBICS supports the signature procedures "Security file" and "Smartcard". MVSC supports both procedures as well as all already existing signature versions (A004 / A005 / A006) and key lengths.

Note
For the use of a smartcard, appropriate hardware is required (smartcard reader / signature cards).

Security file

A function available in MVSC enables the creation of an own password-protected security file. For security reasons, the respective storage locations of the physical security file and its password should not be next to each other, as a certain risk remains in spite of the encrypted storage in MVSC.

Tip
Store your security file on a USB stick.

Caution
With six incorrect password entries, the security file and the access will be finally deleted.

Smartcard

If you have an EBICS-capable smartcard, we recommend using it instead of a security file. Smartcards are generally more secure as the private keys stored on them can never be stored on a hard drive or anything similar. The keys for signatures and encryption on a smartcard are secured with two PINs (card and signature PIN). However, the two PINs may also be identical. The PINs are entered via the keypad of a suitable smartcard reader. This provides an additional level of security against malware attacks (e.g. Trojans) compared to passwords entered via a normal keyboard.

Caution
Should you possess a personalised smartcard, please ensure that you know the PINs of the card. If you have not changed them yet, you will find the initial PINs in your PIN letters.

Note
The signature procedure "Smartcard" is not suited for automated data transfer. If certain access data is to be used in the console mode, you must select the signature procedure "Security file".

1.3. Supported order types

General information

The MVSC application supports all order types of EBICS. However, it only displays the order types that have been set up and/or assigned to the respective access at the EBICS server.

Synchronising order types

Should you required additional order types, they must be activated at the EBICS server. Subsequently, the order types known in MVSC must be synchronised with the bank server system again.

Electronic distributed signature ("EDS")

The "EDS" is a location-independent release system for orders. This means that if user X of a customer is not sufficiently authorised to execute an order type with his signature alone, the order is moved to a so-called "order pool". Other users (Y) that are authorised for this account can now download an overview of the orders collected there. After the overview has been successfully downloaded, user Y can view the orders pending further signatures and sign or cancel them.

1.4. Managing access and connection data

EBICS

To create a functional EBICS access, you need the following information from your BPD sheet:

- Customer ID
- Host name of the EBICS server (8 digits)
- URL of the EBICS server (starting with "https")
- User ID
- EBICS version (supported versions can be downloaded via a button)

After entering and saving this data in the GUI, you must configure the security medium:

- Signature medium
- Signature version (must be known if using an existing medium)
- Path to the security file/card number (button "Assign card")
- If applicable, password for the security file (use in console mode)

Internet

If a direct connection to the Internet is available, no changes to the preset connection type are required. If the connection is to be established via a proxy server, these settings are easy to configure, as well.

1.5. Structure of the program

Menu items

After the dialog user has been successfully authenticated by the application via the login mask, the main window of the application is displayed.

The following figure shows the tabs available to you:

Figure 1.1. Tabs of the main application window

The following table describes each of the tabs:

Tab	Function/description
Data transfers	In this mask, you can perform data transfers. To this end, you select the respective EBICS access and then the desired order type and the files to be transferred. The execution of orders is only possible once your EBICS access data is fully stored in the program and has been initialised at the EBICS bank server. **Note** This tab "data transfers" is only available in the full version of "MVSC". It's not available in "MVSC Sign".
Order History	Here you have the option of displaying the entries from the HAC protocol. You will be shown all orders that relate to the respective customer ID and the respective access ID. **Tip** Under the menu item "Configuration"->"Options" you can specify how long the entries from the order history are displayed. More information on this can be found in the chapter "Check options" in the section "Order History".
Signatures	The EBICS procedure enables the release of orders with the four-eyes principle. The not yet fully authorised orders are then stored at the EBICS bank server and await further signatures by authorised users. Only after one or several further signatures are granted is the order forwarded to the processing systems. The overview of orders pending signatures can then be view in this tab by specifying the EBICS access. The orders can be signed or cancelled.
Accesses	This tab contains all input fields for EBICS access data. The access data is entered here. The accesses tab is also used to create and/ or initialise the related security medium. The order types stored in the system can be downloaded in this mask. If MVSC is to be used in the console mode, you can define the defaults for the call here.
Dialog users	This tab is used to manage the dialog users stored in MVSC. The administrator is authorised to create or delete dialog users and to reset the password of other dialog users. All other dialog users can only change their own password.
Internet	In this tab, you can configure the Internet connection. This setting is valid for all created access IDs. Provided the connection does not use a proxy server, the settings can remain unchanged. Otherwise you must specify the connection data of the proxy server used.
Log	The dialog user's actions are recorded in a log file. A separate file is created for each day. In the tab "Log", you can view and filter these files.

Chapter 2. Technical aspects

2.1. System requirements

Operating system and Java environment

You require a Java runtime environment (JRE) version 1.7 or 1.8 to use MVSC on your computer. Thanks to its flexible structure, MVSC is compatible with all operating systems that support this Java standard (e.g. Windows, Unix, Linux).

2.2. Directory structure

Content of the program directory

After the installation of MVSC, the directory structure depicted in the following figure is located in the specified MVSC program directory:

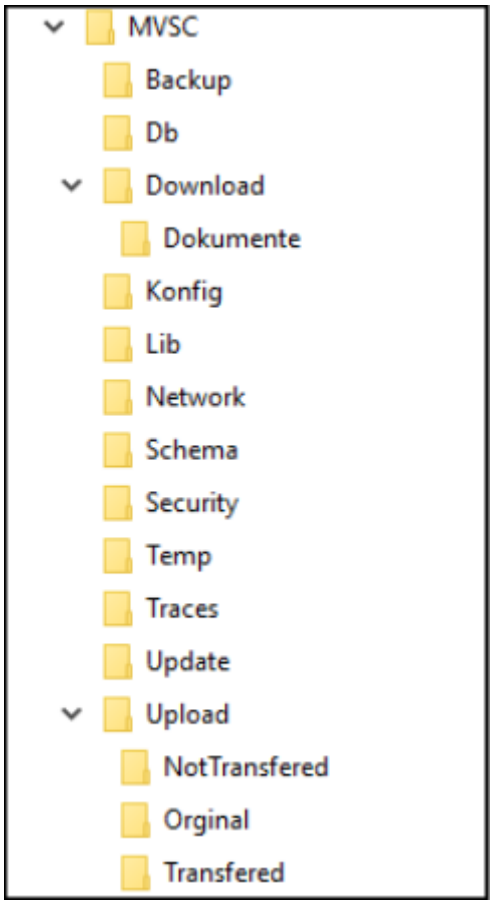


Figure 2.1. MVSC directory structure

Upload and download directories

The directories "Download" and "Upload" are used as the standard inbound directory and standard outbound directory for received files and files to be sent. Both directories can be configured individually per access ID. Below these directories you will find further subdirectories ("Documents" or "NotTransfered", "Original" and "Transfered").

Documents directory

The directory "Dokumente" is used as a storage directory for received protocol files (order types "PTK" and "HAC") and INI letters. To separate documents from order files, the directory is usually located below the specified download path. This directory can also

be configured individually per access ID. For more information, see the chapter "Default settings".

Processed files

The directory "Transfered" below the upload directory serves as storage location for transferred files. This path can also be configured individually per access ID. The distinction between already transferred files and files that are pending transfer is necessary as otherwise the same files could be transferred multiple times in the console mode. If the data transfer fails in the console mode, the affected file is moved to the directory "NotTransfered". In the GUI mode, the directory "NotTransfered" is irrelevant.

The directory "Original" is used to store the original file if the option "Send selected sepa-files as IBAN-Only" is used. More information on this option can be found in the chapter "Default settings" in the section "Group 'Other settings'".

Security files

In the security directory, MVSC stores the generated security files (*.ESK). Additionally, the public keys of the different EBICS bank servers are stored here (*.PKD).

Log and trace files

Technical log files, which can be helpful for us as manufacturer in case of errors, can be found in the directory Traces. As of version 2.0, this directory also contains the daily written action logs.

Configuration files (valid until MVSC version 2.0)

All specified configuration data can be found in the config directory. This includes created access IDs, file filter settings, permitted order types and the Internet connection data.

Database directory (valid as of MVSC version 2.0)

As of MVSC version 2.0, all specified configuration data can be found in the DB directory in an encrypted database. This includes created access IDs, file filter settings, permitted order types and the Internet connection data. When the MVSC version 1.0 is updated to 2.0, the data from the config directory is automatically imported into the database.

Other directories

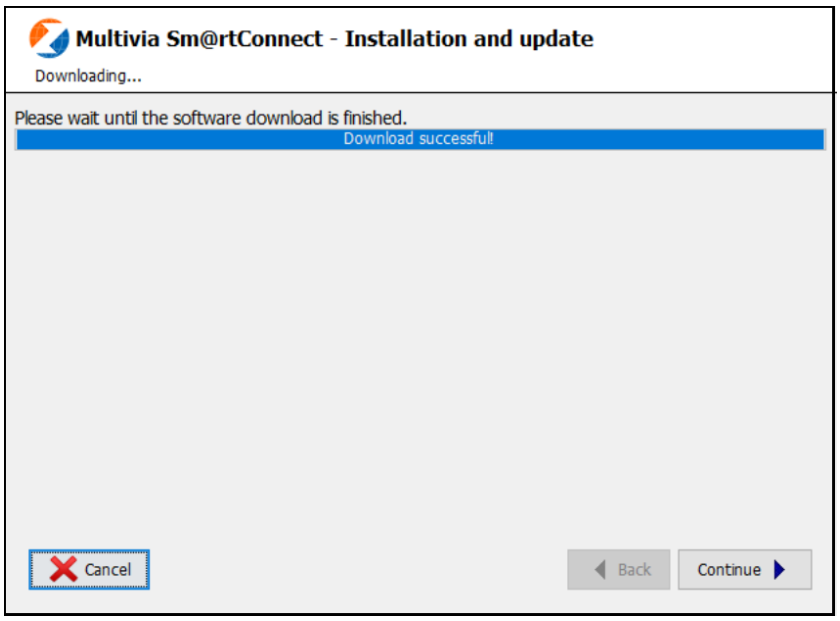
The directories "Lib", "Temp" and "Network" are not relevant for operating the application. However, data transfers via EBICS are not possible without the files in the Lib directory. When the software is updated, the update package to be installed is first stored in the folder "Update". The most important files are backed up in the directory "Backup" prior to the update. The XML schema files (*.XSD) required for the validation of SEPA XML files are located in the directory "Schema".

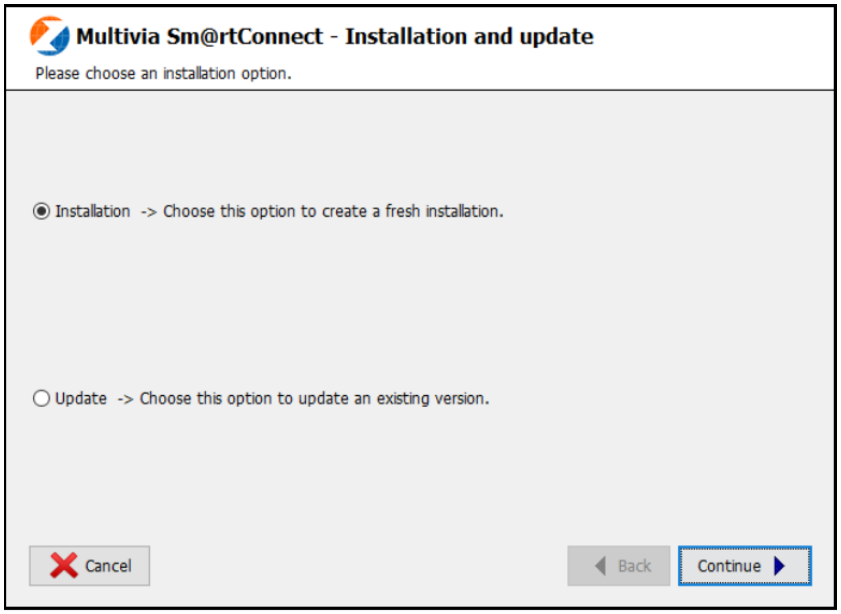
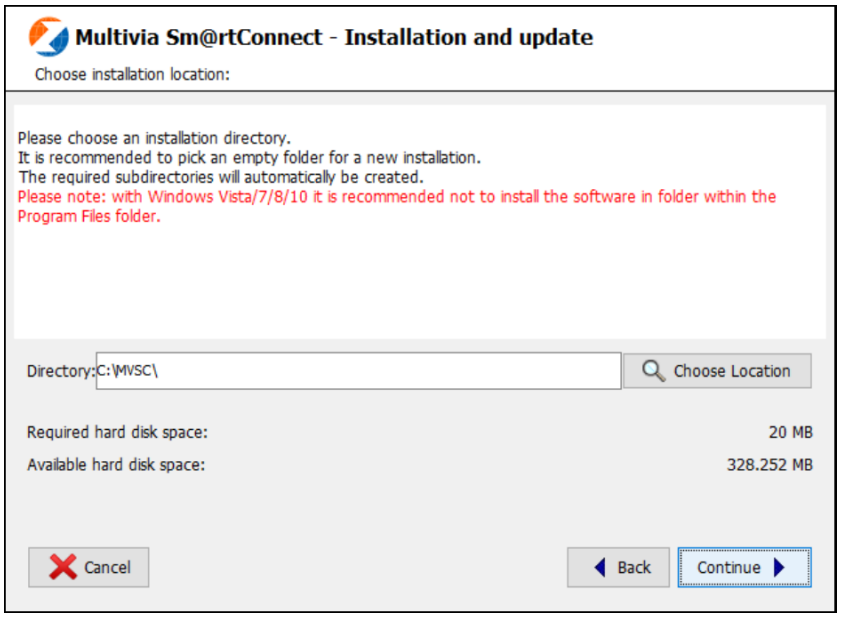
Chapter 3. Installation

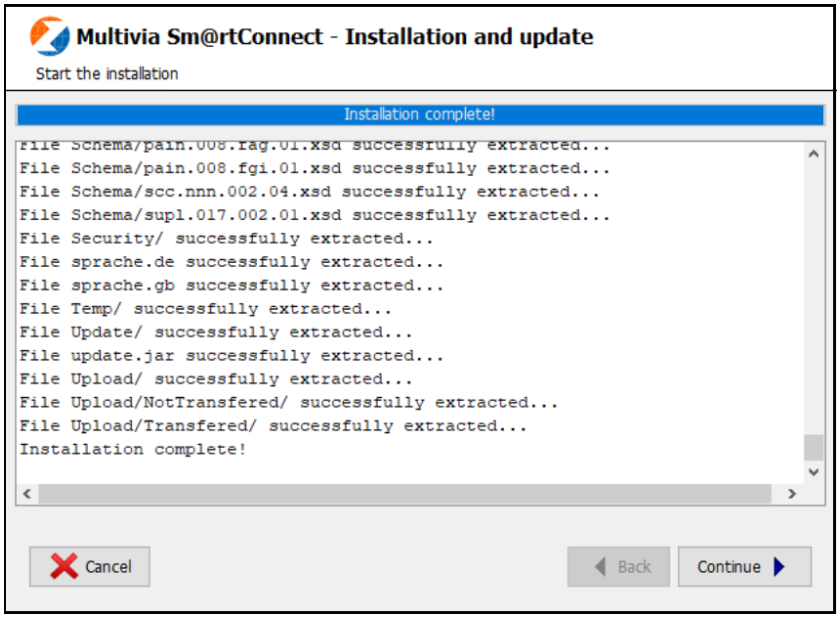
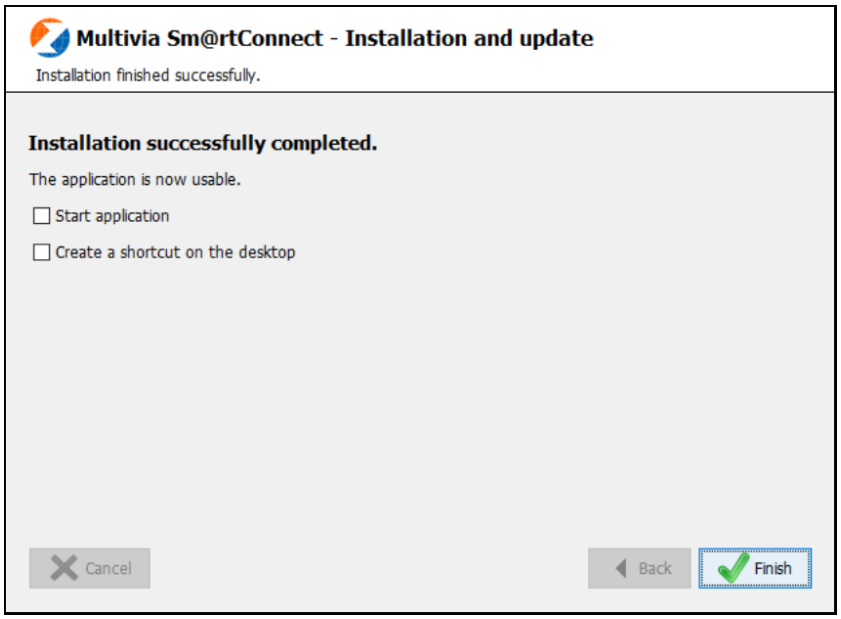
3.1. Installation with the wizard

Installation with the wizard "install.jar"

As of its version 1.0, the application includes an installation wizard that downloads the program from a server and guides you through the installation in just a few steps. This installation wizard "install.jar" is described in the following table:

Step	Procedure
1	First select the language you want to work with. The selected language is then saved as a default value for future calls of the program. Regardless of this, you can still change the language as described in the introduction in the section "Multilingualism". The following mask for language selection is displayed:  **Figure 3.1. Language selection during installation** You then get to the next step via the button "OK".
2	Start the download of the application and wait until the process is finished. The following mask is displayed:  **Figure 3.2. Download of the application** You then get to the next step via the button "Continue".
3	Select whether you want to perform a new installation or whether you want to update an existing version. The following figure shows the selection mask:

Step	Procedure
	 **Figure 3.3. Installation or update of an existing version** Go to the next step via the button "Continue".
4	Select an installation directory. The following figure shows the mask "Select installation directory":  **Figure 3.4. Select installation directory** Go to the next step via the button "Continue".
5	Wait for the installation process to finish. The following mask is displayed:

Step	Procedure
	 **Figure 3.5. Installation process** You then get to the next step via the button "Continue".
6	A confirmation message informs you that the installation process is completed. The following figure shows the confirmation mask:  **Figure 3.6. Installation complete** Exit the installation wizard via the button "Finish".

Update of the signature version of an access ID

As of MVSC version 2.5.0, if you have an access ID with a signature version smaller than "A006", you will be presented with a mask offering to update to signature version "A006". The access IDs that have a smaller signature version than "A006" are listed on this mask.

The following figure shows an example of this mask:

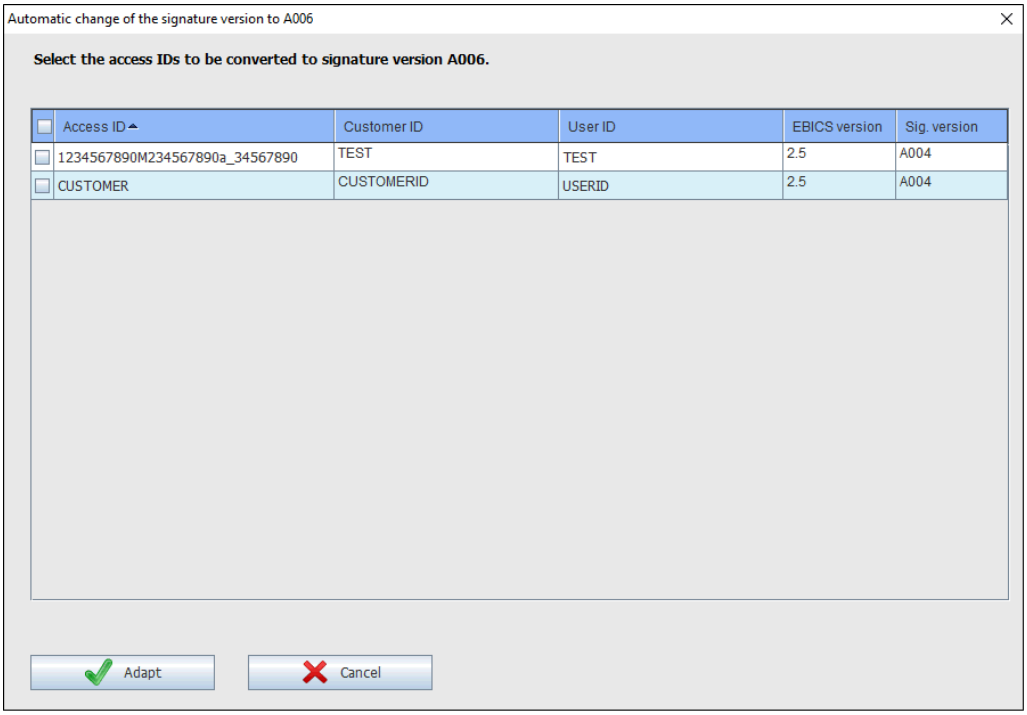


Figure 3.7. Automatic change of the signature version to A006

Here you can first select the access IDs for which you want to run an update to signature version "A006". Use the button "Adapt" to execute the update. In order to change no access IDs select the button "Cancel".

Update

As of the MVSC version 1.0, the application has an integrated update function. After the login, the application automatically checks whether updates are available. If an update is available, a corresponding message is displayed and you can either confirm the update or reject it. The update check can be repeated via the menu "Help->Update".

Chapter 4. Setup

4.1. Application start

Authentication

An MVSC administrator access is used for the login to the application. The initial administrator password is "xxxx". This has to be changed at the first login. The new password has to consist of at least eight characters. It has to consist of letters, numbers and at least one special character.

Then the password can be changed at any time in the tab "Dialog users".

The administrator can create other dialog users who have their own password for application login.

4.2. Prerequisites for EBICS communication

Customer ID at the EBICS bank server

To participate in the EBICS procedure, a customer requires a customer ID that must be set up on the EBICS bank server system. Various user IDs that usually represent the employees of a company can be set up for one customer ID. The authorisations of each employee are stored on the EBICS bank server at the respective user ID and can only be changed there.

BPD sheet

For each user ID, a "BPD sheet" is available that contains, among other data, all information on the user's authorisations. This includes the connection data of the EBICS server (host name/ URL), the permitted order types, the public bank keys to be checked during initialisation (hash values) and a list of accounts for which a user ID has authorisations. To set up an access ID in the tab "Accesses", the "Customer ID", "User ID", "Host name" and "URL address" must be copied to the mask and then saved.

4.3. Specifying an Internet connection

Internet connection

An Internet connection is required for EBICS communication. The Internet connection is specified in the tab "Internet" and centrally stored for all dialog users and all EBICS accesses.

Use without a proxy server

If the Internet connection is not to be established via a proxy server, there is no need to adjust the data in the mask "Manage Internet connection".

The following figure shows the mask "Manage Internet connection":

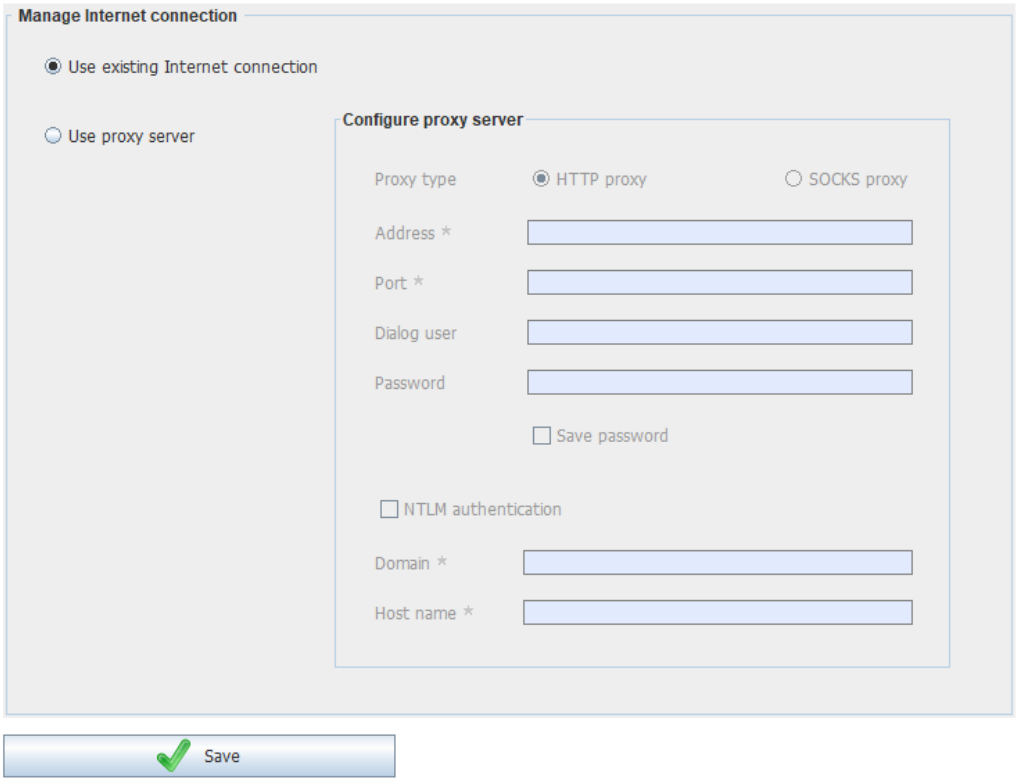


Figure 4.1. Internet use without proxy

Use of a proxy server

If a proxy server is required to establish a connection with the Internet, at least the address (or IP) and port of the server must be entered in the appropriate fields. As some proxy servers demand an authentication, the dialog user name and password can also be specified in the application.

Password for the proxy server

If the password for the proxy server shall not be stored in the application, it can also be entered via the GUI during a data transfer. In console mode, however, the password for the proxy server (if it is required) must also be stored.

Use of the NTLM authentication

"NTLM" is an authentication method developed by Microsoft for computer networks. It ensures the authentication within a domain via the name of the respective workstation. To use this kind of authentication, you must specify the domain of the addressed proxy server as well as the name of your own workstation (host name).

Note
The NTLM authentication can only be used in suitably set up networks. In most networks, the direct authentication via a specified proxy server is sufficient.

4.4. Specifying EBICS access data

Menu item

The EBICS access data is specified in the tab "Accesses". The setup of a functional EBICS access in MVSC is performed in several steps:

Step	Procedure
1	Copy the EBICS access data from the BPD sheet to the input screen and save your input via the button "Save".

Step	Procedure
2	Set up a new security medium via the button "Generate new" or assign an already existing security medium.
3	Initialise the specified security medium at the EBICS bank server (disclosure of your own public keys). **Caution** As a result of this step, you will receive a so-called "initialisation letter". Submit this initialisation letter to your bank in written form so that the bank can activate your keys. The next step can only be performed after the activation.
4	Download your authorisations. **Note** During this process, the server certificate and the public keys of the bank are downloaded, if applicable.

Via the menu item "Configuration -> Set up access ID", you can start a wizard that helps you create an access ID. The following information is then requested step by step.

Selecting the card reader

If you want to use a smartcard as signature medium for your EBICS access, you must select a standard smartcard reader. You can change the standard smartcard reader at any time via the menu "Configuration -> Select card reader". The standard smartcard reader selected from the list will be used for all future accesses to the smartcard.

If no standard smartcard reader has been selected yet, or if the specified standard smartcard reader was not found, MVSC offers you a selection list.

Identification of the access data

The access data entered is always identified via the specified access ID during the future use of the application.

Caution
The access ID itself can no longer be changed after its setup.

Setup process

The following table describes the setup of a new access ID:

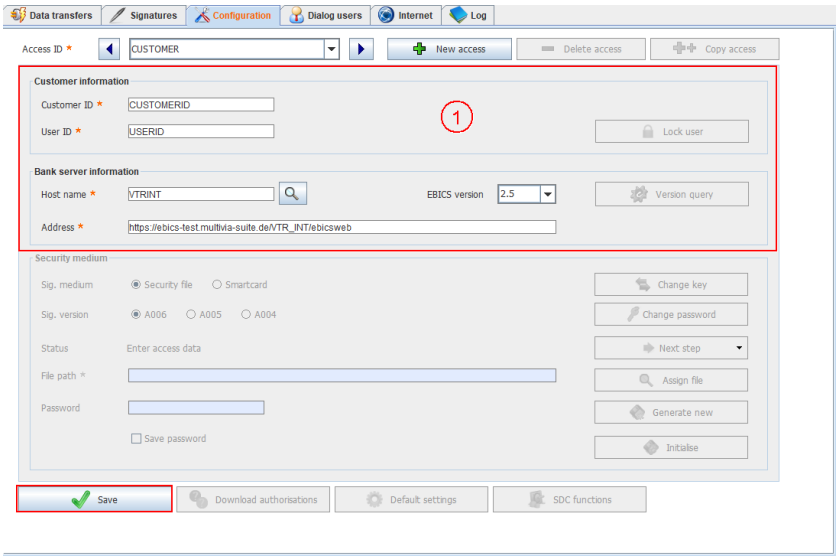
In the second step, a distinction is made between the three signature media

- certificate
- security file
- smartcard.

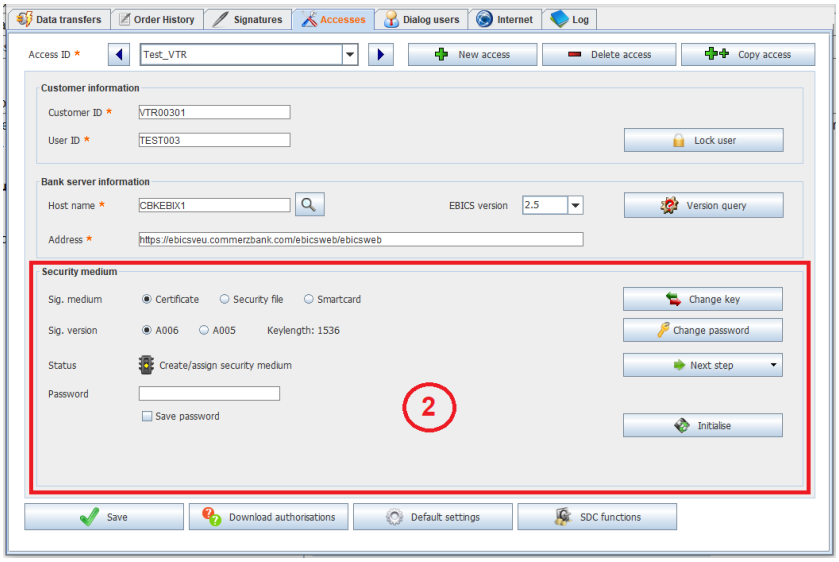
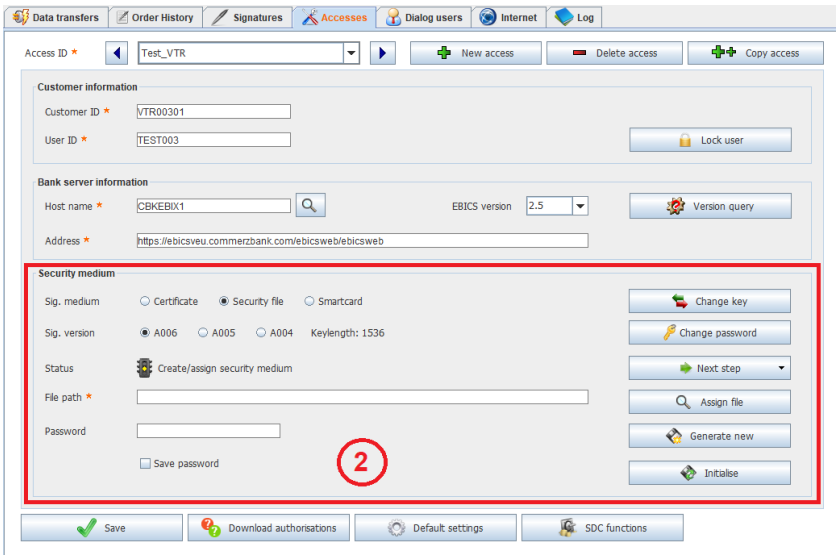
These three alternatives are described in steps 2a, 2b and 2c. So only one of the three steps 2a, 2b or 2c has to be carried out.

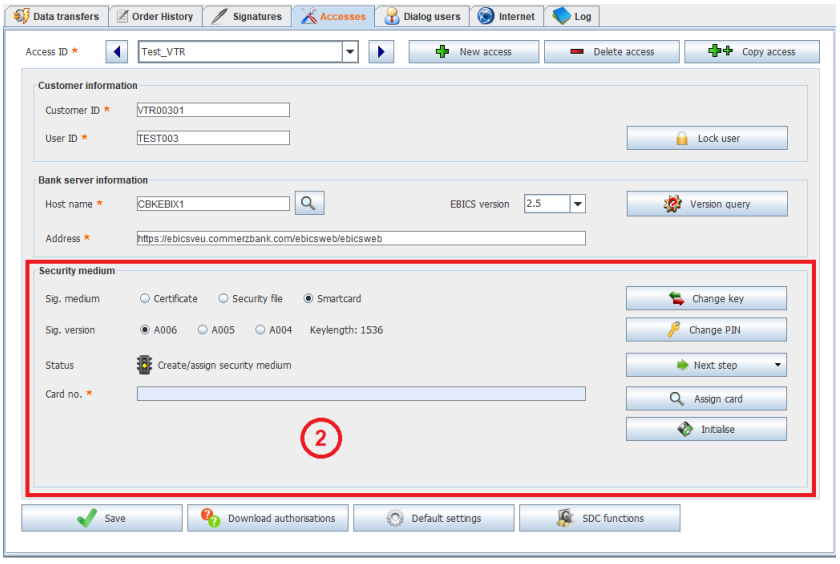
Tip
It is recommended to use the certificate.

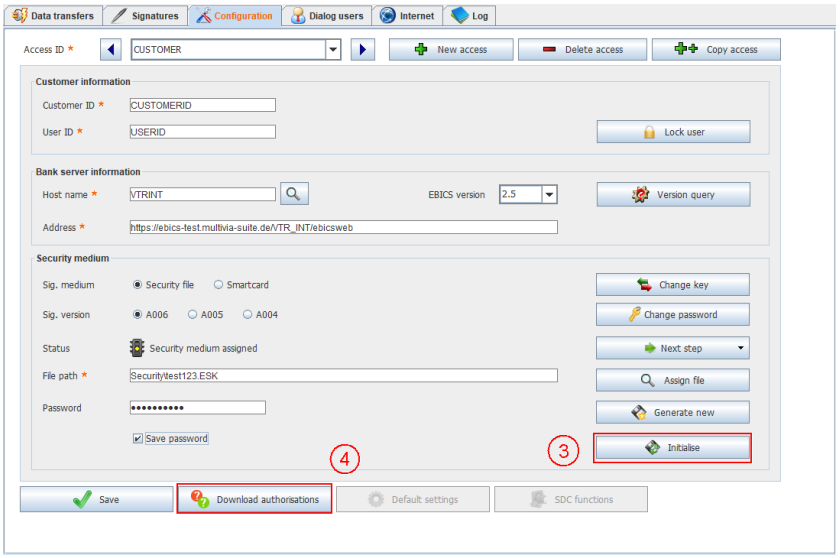
Caution
From the EBICS version "EBICS 3.0" no new chip cards and security files can be generated or initialized. However, existing chip cards or security files can still be used.

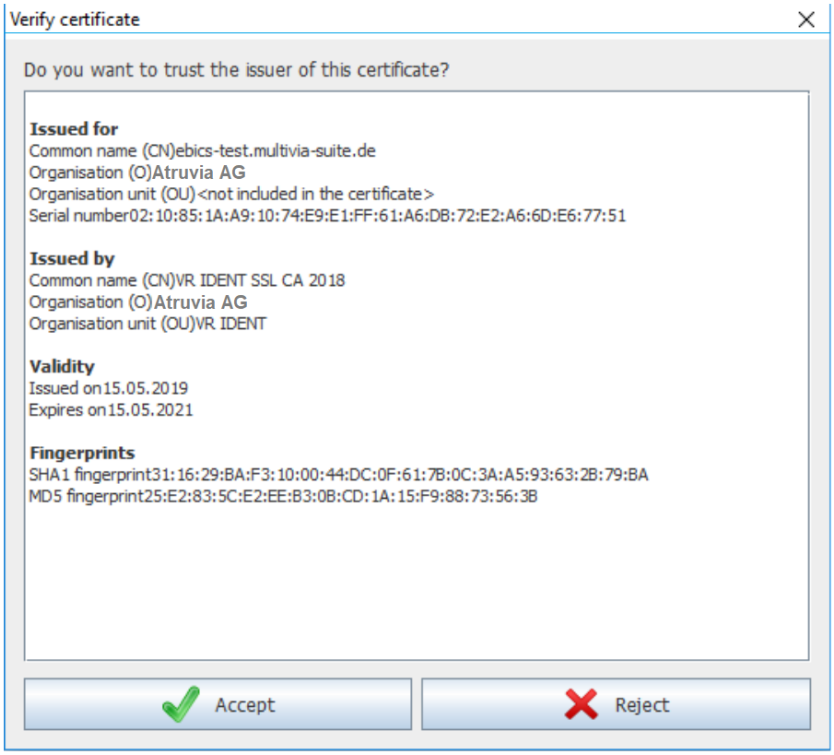
Step	Procedure
1	Copy your EBICS access data from the BPD sheet to the configuration mask. Basically you can choose any name for the access ID. The length may only be a maximum of 30 characters. Only the following characters may be used: • Upper and lowercase letter (A - Z, a - z) • Umlauts (Ä, ä, Ü, ü, Ö, ö) • Digits (0 - 9) • Underscore **Note** Old accesses are unaffected by this restriction. The following figure shows the configuration mask:  **Figure 4.2. Data entry in Multivia Sm@rtConnect, step 1** Enter the data specified in the BPD sheet and define an access ID of your choice in the upper area of the mask. The following fields must be filled: Select your host name and the EBICS version. Save your input via the button "Save". Depending on the signature medium, continue with step 2a, 2b or 2c: • Step 2a: Certificate • Step 2b: Security file • Step 2c: Smartcard
2a	After the data from step 1 is saved, the security medium must be configured. This example uses a certificate as medium. **Signature medium certificate:** The following figure shows the mask for configuring the security medium using a certificate as example:

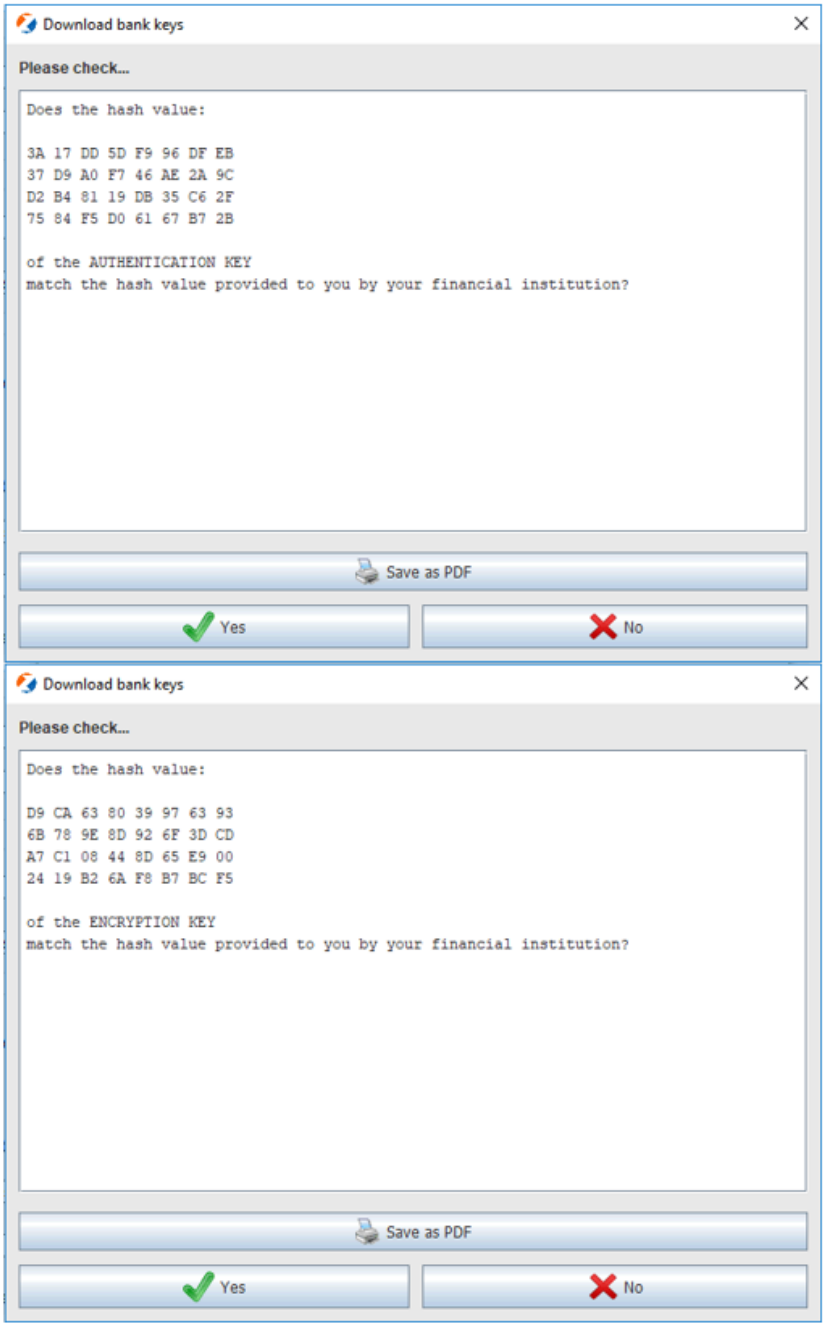
Field name in MVSC	Field name in BPD sheet
Customer ID	Customer ID (page 1)
User ID	User ID (page 2)
Host name	EBICS host IDs (page 1)
Address	Bank parameter URL / EBICS URL (page 1)

Step	Procedure
	 **Figure 4.3. Data entry in Multivia Sm@rtConnect, step 2a** **Prerequisites for using a certificate:** • Germany: At least EBICS version 2.5 is used. • Europe except Germany: At least EBICS version 3.0 is used. **Procedure when using a certificate:** • Select "Certificate" as signature medium. • Select which signature version your certificate corresponds to (A006 / A005). • Create a password for your certificate. • Repeat the password you specified. • Optional: store your password for the generated security file (checkbox "Save password").
2b	After the data from step 1 is saved, the security medium must be configured. This example uses a security file as medium. **Signature medium security file:** The following figure shows the mask for configuring the security medium using a security file as example:  **Figure 4.4. Data entry in Multivia Sm@rtConnect, step 2b**

Step	Procedure
	Procedure when using an existing security file: • Specify which signature version matches the keys in the target file (A006 / A005 / A004). • Click on the button "Assign file". • Select the already existing security file and confirm the selection dialog. • Save your access data via the button "Save access". **Procedure for generating a new security file (no longer possible from EBICS 3.0):** • Specify which signature version shall be generate with the security file (A006 / A005 / A004). • Click on the button "Generate new". • Specify the path and/or file name under which the security file shall be stored. • Create a password for your security file according to the password rules displayed. • Repeat the password you specified. • Optional: store your password for the generated security file (checkbox "Save password").
2c	After the data from step 1 is saved, the security medium must be configured. This example uses a smartcard as medium. **Signature medium smartcard:** The following figure shows the mask for configuring the security medium using a smartcard as example:  **Figure 4.5. Data entry in Multivia Sm@rtConnect, step 2c** **Prerequisites for using the smartcard** • A smartcard reader is connected and installed on the computer. • The standard smartcard reader has been selected. • An EBICS-capable smartcard (card of the type "SECCOS 6") is used. • The initial PINs are known and/or have already been changed. **Procedure for assigning a new smartcard (no longer possible from EBICS 3.0):** • Select "Smartcard" as signature medium. • Select which signature version matches the keys on your smartcard (A006 / A005 / A004). • Click on the button "Assign card".
3	All data for the EBICS access is now specified and the security medium can be initialised at the bank server system.

Step	Procedure
	The following figure shows the procedure:  **Figure 4.6. Data entry in Multivia Sm@rtConnect, step 3** **Procedure for initialising the security medium:** First, click on the button "Initialise". In this process, the public keys of your security medium are transmitted to the EBICS bank server and matched with the access data (user ID) stored there. As an "acknowledgement", you receive a so-called initialisation letter, which is stored in the installation directory under "Download/ Documents" as a PDF file. By signing the INI letter, you confirm your own signature medium. The INI letter must be sent via an independent transport channel (e.g. via fax) to the operator of the EBICS bank server. By means of the hash values printed on it, the user ID can be activated at the bank server system. Once this step is done, the EBICS access (or the access ID in MVSC) is fully initialised and functional. **Note** The SSL certificate (for the https connection) of the respective EBICS bank server may yet be unknown to MVSC. In that case, the application downloads the certificate of the server and displays information on the issuer of the certificate. If the issuer is trustworthy, the certificate can be accepted; otherwise, it should be rejected. Without an accepted SSL certificate, no EBICS communication is possible. **Note** It may occur that a certificate cannot be imported as it was not issued by an official trust centre (e.g. "Verisign"). If that is the case, communication with the corresponding bank server system is not possible. Subsequently, the initialisation process can be completed in MVSC.
4	Appropriate SSL certificates are required for EBICS communication via the HTTPS protocol. If the certificate is not available yet, it is downloaded in this step. Often the required certificates are already stored in the system and this step is thus no longer necessary. After the certificate has been downloaded, the information required for you to verify the certificate is displayed. The following figure shows an exemplary display:

Step	Procedure
	 **Figure 4.7. Verifying the certificate** **Caution** It must be apparent in the displayed information that the certificate originates from the server configured in the access data. For example, the value "Issued for" must contain part of the EBICS address / URL copied from the BPD sheet.
5	To execute the order types permitted for the access ID, they must first be downloaded from the bank server system and stored in MVSC. To do that, click on the button "Download authorisations". Should this be your first transaction with the respective bank server, MVSC automatically downloads the public keys of the bank server system during the process. When confirming the public keys, however, you should note the following: • The keys are displayed in a processed form and have to be verified by you. The following figure shows the display:

Step	Procedure
	 **Figure 4.8. Download of the bank key** • Compare the displayed hash values with the ones printed on your BPD sheet on page 2. • If the values match, confirm both keys with the button "Yes". If the values do not match, the keys must not be confirmed or saved for security reasons (suspected manipulation). Once the public keys have been confirmed, the download of the authorisations resumes automatically.
6	Once the authorisations have been downloaded, the list of the executable order types is also synchronised with the EBICS bank server. If, for example, an order was newly assigned at the EBICS bank server, it can be executed in MVSC after the authorisations have been synchronised. Additionally, you can view the accounts for which the respective access is authorised to submit payment files.

Step	Procedure
	The following figure shows the mask:

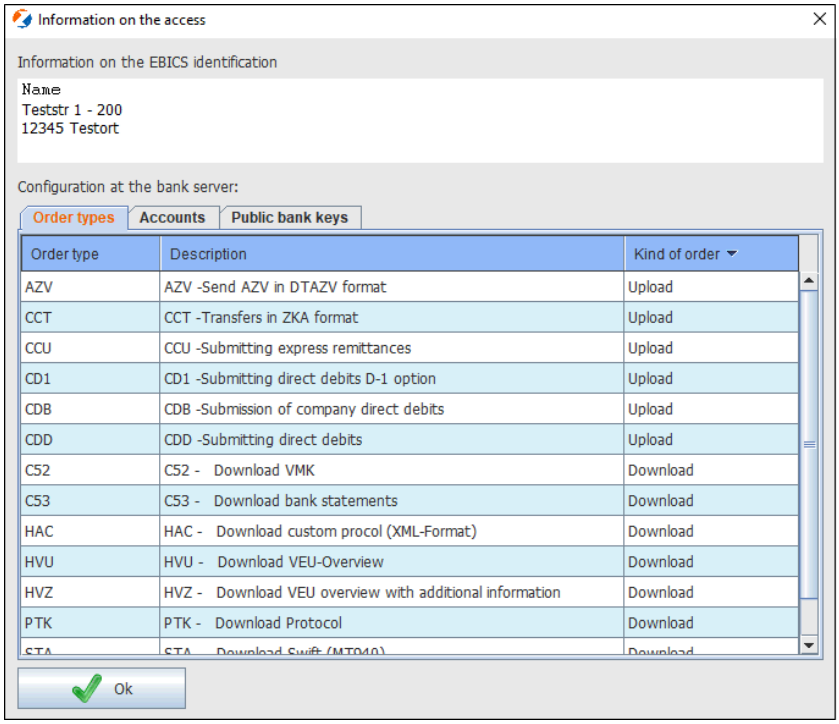


Figure 4.9. Information on the access

After the list of order types has been synchronised for the first time, click on the button "Default settings". In the following dialog, you can define the default directory paths in the tab "Data transfers". The post-processing of sent files can also be configured here. For more information, see the chapter "Default settings".

Change key

You can use the function "Change key" for changing your user keys at the bank server.

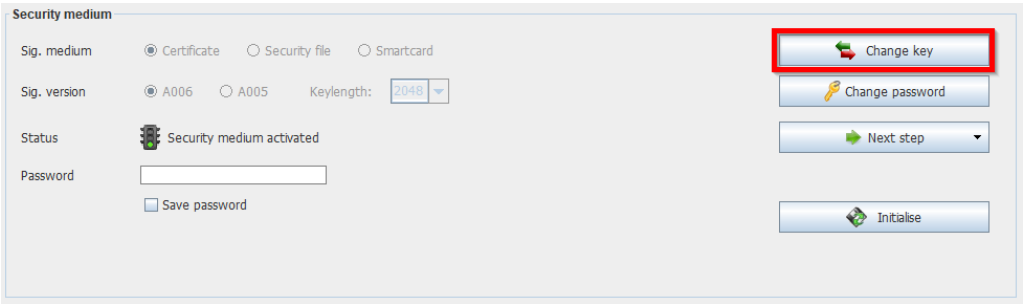


Figure 4.10. Call up the function "Change key"

In the following dialog, select your signature medium, your signature version and the keylength.

This is shown as an example in the following three figures:

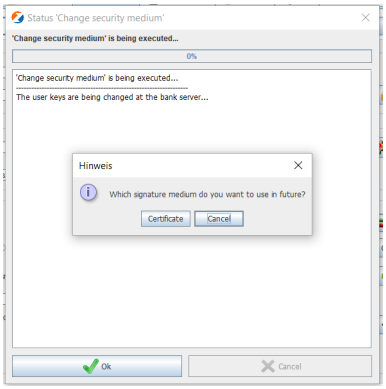


Figure 4.11. Key change - selection of the security medium

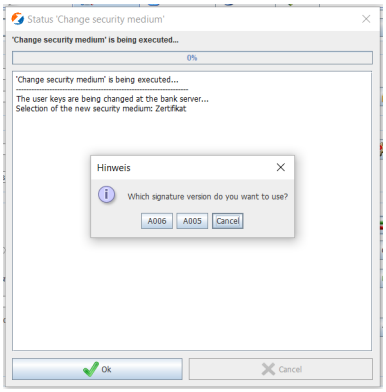


Figure 4.12. Key change - selection of the signature version

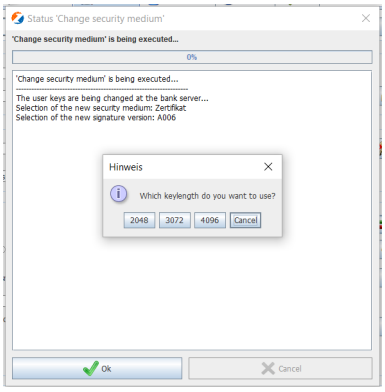


Figure 4.13. Key change - selection of the keylength

4.5. Importing access data

Import from existing installation

If an MVSC version is already installed, the configuration data stored therein can be imported. For a successful data import, the source installation directory must be specified. Additionally, the administrator password of the source installation must be entered before the import process is started. The data import can be initiated via the menu item "File->Import".

Note

If the dialog users of the application shall be migrated from the source installation, the password of the administrator is also stored in the target installation.

Import process

To import data from another MVSC installation, proceed as follows:

Step	Procedure
1	Enter the directory of the source installation. This directory should contain files such as "MVSC.jar" and "defaults.xml".
2	Select the data you wish to import from the source installation: • Access IDs (All EBICS configuration data from the tab "Accesses" is migrated to the target installation.) • Dialog users (All dialog users are migrated to the target installation with their passwords; the administrator password is also migrated.) • Internet settings (The Internet connection data from the tab "Internet"is migrated to the target installation and, if applicable, overwritten there.)
3	Start the import process via the button "Import".
4	Enter the administrator password of the source installation.
5	Check the imported data. **Note** During the import of access IDs, the directory paths configured under "Default settings" may be adjusted to the target installation. The paths should be checked and, if required, corrected after the import process is finished. **Note** If an access ID to be imported already exists in the target installation, a new name (access ID) must be specified for this EBICS configuration.

The following figure shows the file import mask:

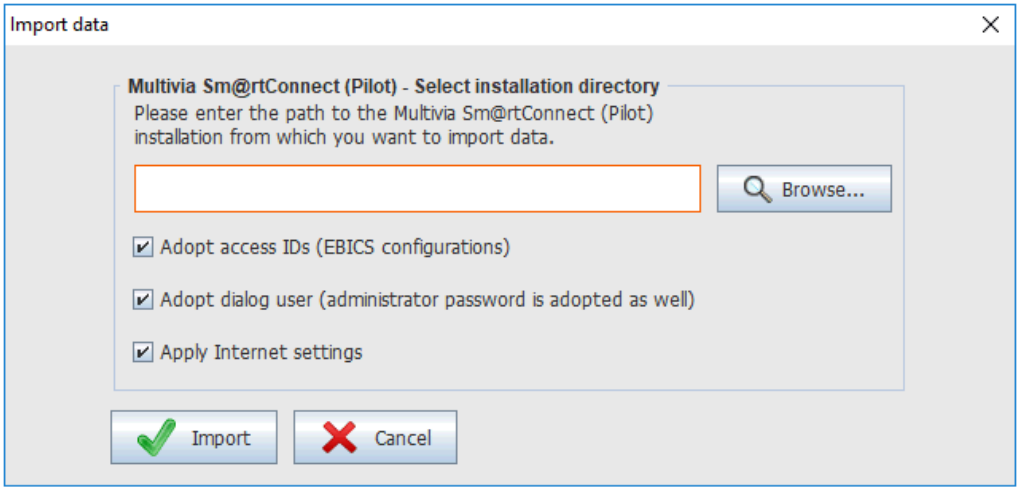


Figure 4.14. File import of data in case of a new MVSC installation

Check

After the import activities are completed, you can check in the so-called "Import log" which data has been imported and whether any errors occurred during the process. It is also recommended to check the following:

- Are all path settings in the "Default settings" still correct?
- Do passwords for security files have to be stored anew? (console call)
- Are any more documents from the old installation required? (e.g. INI letters)

4.6. Licence server

Licence key management

Each MVSC software bundle must be registered by means of a licence key. The validity of the licence key is checked during the login to the "MVSC" application. At this point, it is checked whether the application is already registered. If it is not properly registered, a message displayed during login to the application informs you how long it may still be used without registration. Generally, the application may be used unregistered for 60 days without restrictions. Afterwards, the use of "MVSC" is limited. File transfers can no longer be performed.

The following figure shows the login mask without registration:

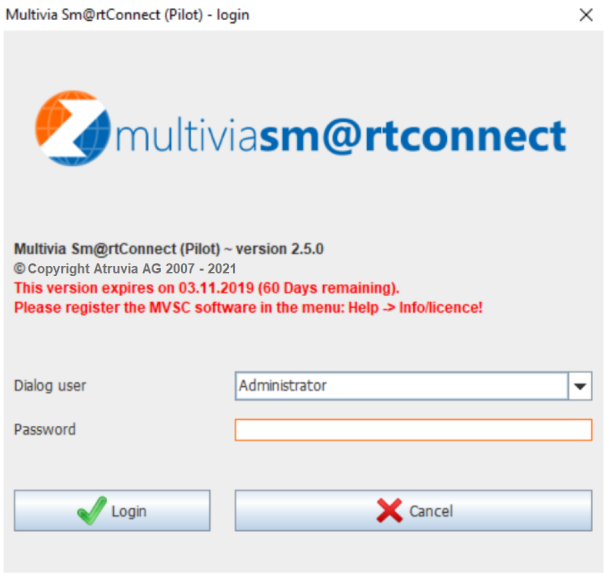


Figure 4.15. Login without proper registration

The following figure shows the login mask after registration:

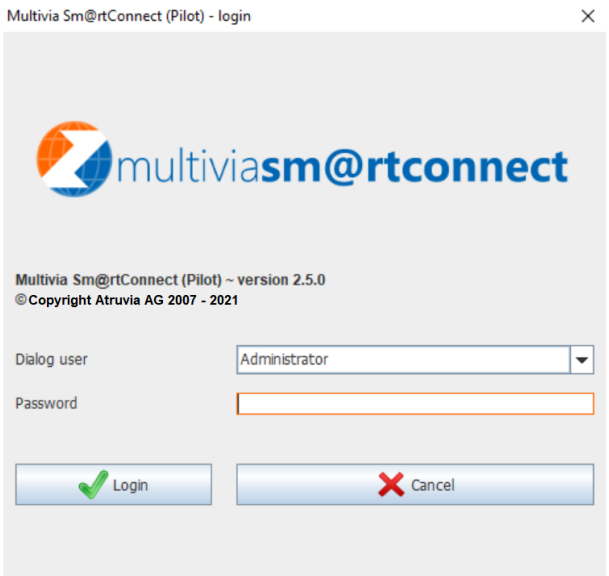


Figure 4.16. Login with proper registration

Licence key registration

The registration at the licence server is performed in the MVSC application via the menu item "Info/licence" in the menu "Help".

The following figure shows the menu "Help":

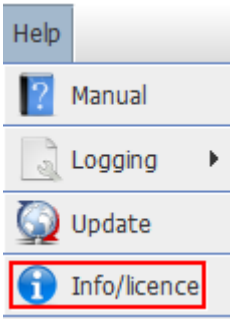


Figure 4.17. Info/licence

Once you have entered the licence key, you can immediately activate the "MVSC" application via the button "Register & check". If the registration is successful, a confirmation message is displayed. The dialog user can check the status of the licence key at any time.

The following figure shows the mask for entering the licence key:

Figure 4.18. Register and check licence key

The following figure shows the confirmation message of the registration:

Figure 4.19. Licence key successfully registered

Chapter 5. Using MVSC

5.1. General information

Possible uses

To fulfil the prerequisites for using MVSC, you should first set up the various accesses (Internet/EBICS). After this is done, you can transfer data directly via the GUI. For use without the GUI, however, additional prerequisites must be fulfilled.

5.2. Data transfer via the GUI

5.2.1. Sending files

Executing an upload order type

To transfer files via EBICS in MVSC, go to the tab "Data transfers". There you can initiate a data transfer in just a few steps, provided the configuration settings were specified correctly.

The following figure shows an exemplary mask for performing a data transfer (file upload):

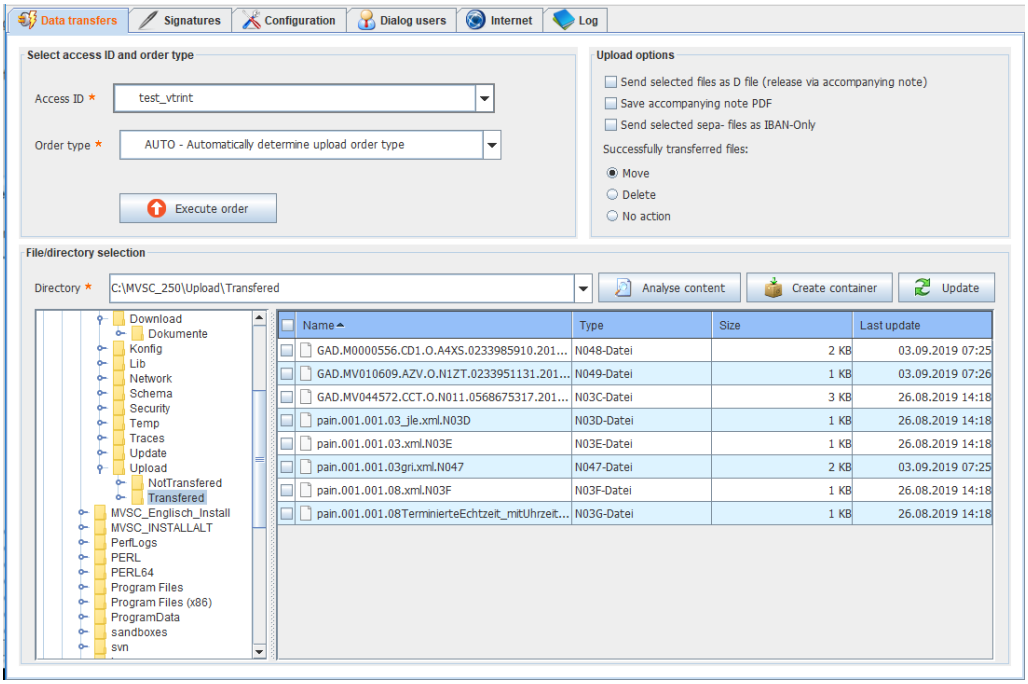


Figure 5.1. File upload

To perform a file transfer, proceed as follows:

1. Select the access ID (EBICS configuration) that shall be used to transfer the data.
2. Specify which order type the file(s) shall be sent with.
3. Select the files to be transferred via the corresponding checkboxes within the table.
4. Click on the button "Execute order" to transfer the selected files.

Tip
In the selection of the order type 'AUTO' all upload files from the upload directory by the determined order type will be submitted to the bank server.

Order types and file formats

When sending files, the order format in the file to be transferred (e.g. "SEPA") must match the selected order type. For example, the order type "CCT - Send SEPA credit

transfer file" should only be used to transfer XML files. If another format is sent with this order type, the EBICS bank server accepts the order but does not process it further. The relevant information can be found in the Customer protocol (order types "PTK" and "HAC").

Tip
The function "Analyse contents" provides information on the respective order format and may thus facilitate the selection of the order type.

After the data transfer

Once the data transfer is finished, the result of the transfer is displayed for you to check. The following figure shows an exemplary display:

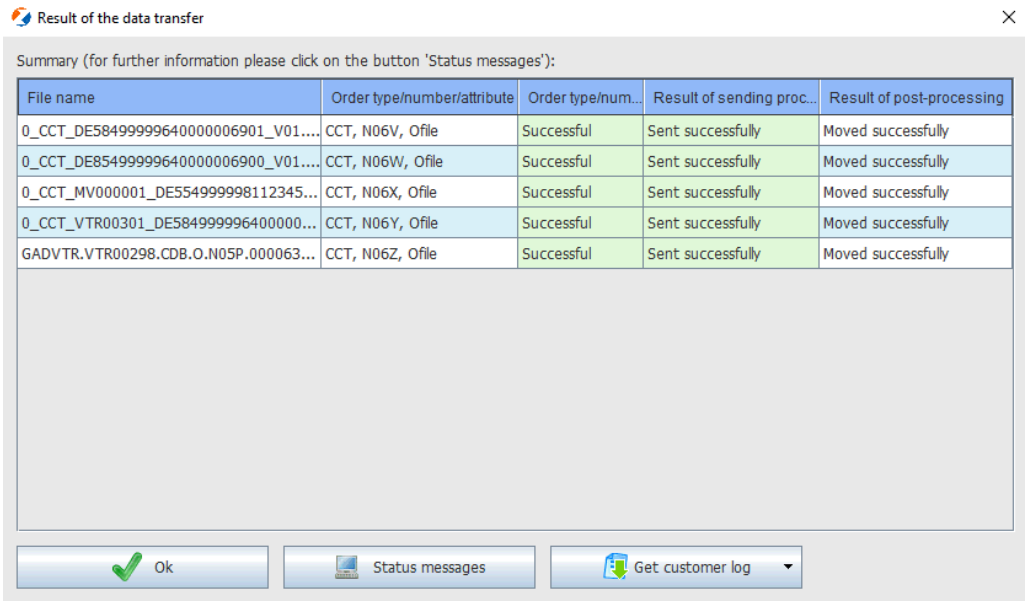


Figure 5.2. Result of the data transfer

By means of the button "Status messages", you can view the status information displayed during the transfer process. The following explains the meaning of the individual table columns:

Column	Content
File name	Name of the transferred file
Order type/number/attribute	The values indicate which order type was used to send the file and which order number was allocated during the data transfer. Additionally, it is indicated which order attribute the file was sent with: • "O file": with electronic signature • "D file": without electronic signature • "Release via accompanying note"
File validation	The value indicates whether the file validation was successful. • "Successful": the order format has been checked and no errors were found. • "Not successful": the order format has been checked and errors were found.
Result of sending process	This column indicates the result of the EBICS data transfer.

Column	Content
	Note The information relates to the sending process only, not to the further processing on the server.
Result of post-processing	Depending on the settings in the data transfer mask under "Upload options", this column contains the result of the local post-processing. **Tip** The target directory of moved files is the "Transfered" directory specified at the access ID under "Default settings".

As described, this dialog only provides information on the technical result of a data transfer. To find out whether the sent files have actually been processed, you must download the respective EBICS customer protocol. More information on this can be found in the chapter "Check options".

5.2.2. Downloading files

Provision of download data

In the same way that files can be transferred to the EBICS bank server, files or information can also be downloaded from the respective EBICS bank server. Different download order types are available.

For data to be downloaded, that data must have been provided beforehand at the EBICS bank server for the respective order type.

Example:

Payment orders with different order types have been sent to the EBICS bank server. They were processed successfully and booked on the respective accounts. Subsequently, the booked transactions are then prepared in the format of an electronic account statement (MT94x, CAMT) and made available as a download on the EBICS bank server. The provided transaction data can be downloaded via the respective order type.

Note
It is possible that no data was provided for a certain download order type. In this specific example, this would be the case if no transactions were booked on the accounts of the customer within the previous day.

Executing a download order type

To execute a download order type, go to the tab "Data transfers". Select the access ID that shall be used to download the data.

Then select the download order type and specify the storage location for the received data. You select the desired target directory via the file tree.

Note
For certain order types, the storage location is always the "Documents" directory configured at the access ID, so that received documents are stored as completely as possible in one directory. This is described in the section "Storage location for documents".

The following figure shows the tab "Data transfers":

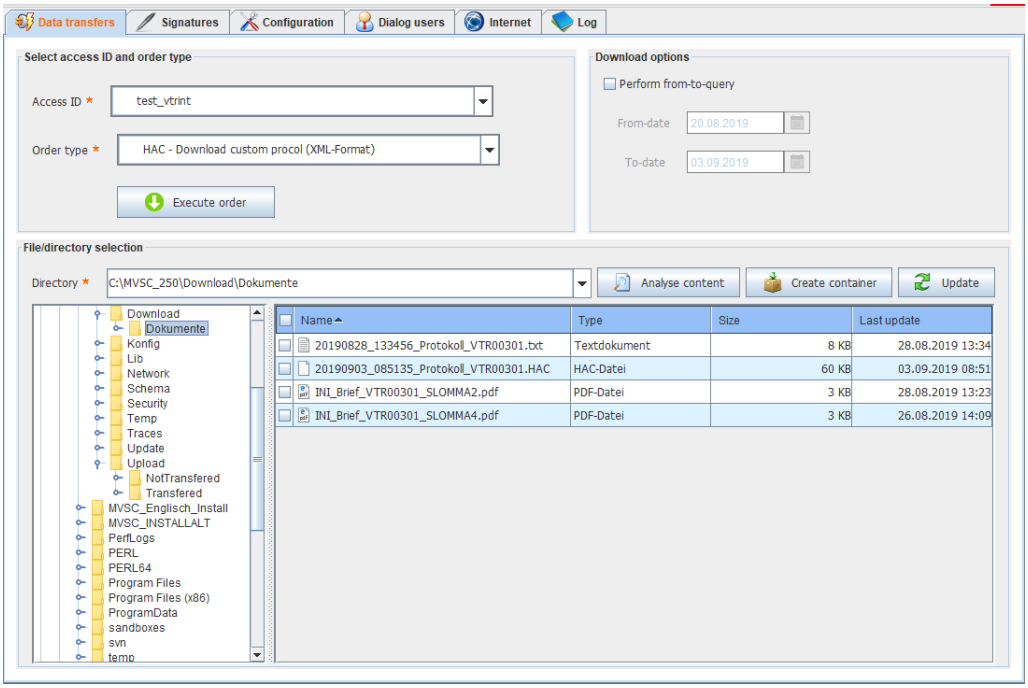


Figure 5.3. File download

Downloading historical data

When download data provided at the EBICS bank server is downloaded for the first time, it is set to the status "Downloaded". They are then no longer available for direct download.

To download already downloaded data again, a so-called "From-to-query" must be performed. To this end, the checkbox "From-to-query" can be activated before the execution of a download order type. If the checkbox is activated, the calendar section below it can be used to define the specific time period for the data download. By means of the button "Execute order" the query is sent to the respective EBICS bank server. The server compiles the data from the requested time period and returns the data found for the order type.

Note

How long certain information remains downloadable via a from-to-query can vary depending on the EBICS bank server. This means that even a from-to-query may not return any information for the specified time period.

Storage location for documents

For each access ID, the storage location for documents is the "Documents" directory specified under "Default settings". The following documents are stored in the directory configured here:

- INI letter: the letter contains the key information exchanged during the initialisation process.
- Customer protocols: these are protocols downloaded from the EBICS bank server and containing information on the processing of sent EBICS orders. Such protocols can be downloaded via two different order types:
  - Order type "PTK": this order type provides the information in a formatted and processed format which, for example, can be read with a simple text editor.
  - Order type "HAC": this order type contains the information in an XML format which is machine-readable and can be evaluated and processed automatically.

Caution

For the order types "PTK" and "HAC", the storage location specified in the data transfer mask is ignored. The data is generally stored in the "Documents" directory of the respectively selected access ID.

5.2.3. Electronic distributed signature (henceforth referred to as "EDS")

Order types and signature classes

The EBICS procedure offers the possibility to release order files via the four-eyes or multiple-eyes principle. This procedure can only be used in combination with upload order types as no signature classes can be assigned to download order types.

The following signature classes can be assigned at the bank server:

Signature class	Description
T (transport signature)	This signature class is only used to secure the data transfer. It cannot be used to release orders.
A (first signature)	This signature class enables the release of orders using the four-eyes principle. The user may only co-sign with users of the signature classes E, A or B.
B (second signature)	This signature class enables the release of orders using the four-eyes principle. The user may only co-sign with users of the signature classes E or A.
E (single signature)	With this signature class, only one signature is required to fully release an order.

As the signature class is assigned per user on order type level, it is possible for a user to have different signature classes for different order types. For example, a user may be authorised to release SEPA credit transfers with one single signature (E signature authorisation for the order type "CCT"), but have to release cross-border orders using the four-eyes principle (A or B signature authorisation for the order type "AZV").

Example for signature levels

The following is an example of signature levels that demonstrates the possible roles pertaining to the EDS:

- The user "BUCHHALT" is an accounting employee in a company. His role is the creation and transfer of order files. Other employees are responsible for the release of orders.
- The user "FREIGEB1" can be a leading employee who is authorised to co-sign an order with another employee using the four-eyes principle to release the order.
- The user "FREIGEB2" can sign orders of FREIGEB1. This means that he could authorise the order together with the leading employee.
- The user "CHEFBOSS" in this example of role assignment would be the CEO of the company, who can release an order with his signature alone at any time.

Table 5.1. Explanation of EDS-related roles

User ID	Order type	Signature class	Possible combinations
BUCHHALT	CCT	T	**Data transfer** The user is authorised to transfer SEPA credit transfer files but is not authorised to release them. All SEPA credit transfer files that are transferred by this user are moved to the signature folder first. The release requires either a single signature (E) or 2 users with A and/or B signature authorisations.
FREIGEB1	CCT	A	**Data transfer** The user is authorised to transfer SEPA credit transfer files with an A signature. These SEPA credit transfer files submitted by this user are already signed with an A signature. They must, however, wait for additional signatures in the signature folder. Only one more signature of the class A, B or E is required to fully release the order.

User ID	Order type	Signature class	Possible combinations
			Electronic distributed signature (EDS) This user can also add his A signature to SEPA credit transfer files of other users that are pending further signatures. This applies, for example, to SEPA credit transfer files that were submitted by the user "BUCHHALT" or "FREIGEB2".
FREIGEB2	CCT	B	**Data transfer** The user is authorised to transfer SEPA credit transfer files with a B signature. These SEPA credit transfer files submitted by this user are already signed with a B signature. They must, however, wait for additional signatures in the signature folder. Only one further signature is required for release, but it must have the signature class A (user "FREIGEB1") or E (user "CHEFBOSS"). A release by two B signatures is not possible. **Electronic distributed signature (EDS)** This user can also add his B signature to SEPA credit transfer files that are pending further signatures. This applies, for example, to SEPA credit transfer files that were submitted by the user "BUCHHALT" or the user "FREIGEB1". However, SEPA credit transfer files that were previously only signed with a B signature cannot be released by this user.
CHEFBOSS	CCT	E	**Data transfer** SEPA credit transfer files submitted by this user are immediately fully authorised and released upon submission. These SEPA credit transfer files thus never wait for additional signatures in the signature folder, but go directly to processing. **Electronic distributed signature (EDS)** All SEPA credit transfer files that are waiting for additional signatures can be immediately released by this user. In this case, it does not matter whether the SEPA credit transfer file was submitted by the user "BUCHHALT" (signature class T), the user "FREIGEB1" (signature class A) or the user "FREIGEB2" (signature class B).

Downloading and editing the EDS overview

To be able to provide further signatures in the electronic distributed signature (EDS) procedure, an overview of the orders waiting for a signature must first be retrieved from the EBICS bank server. You can use the tab "Signatures" to do this.

After specifying the access ID, click on the button "Download overview".

Once the overview is successfully downloaded, further functions become available if the overview actually contains orders. The following table describes the functions:

Function	Description
Cancel	The orders selected via the checkboxes are cancelled. They can no longer be signed by any users. **Caution** If an order was mistakenly cancelled, it must be newly submitted.
Sign	You sign the orders selected via the checkboxes.
Display accompanying note	The order file's accompanying note prepared by the EBICS bank server is displayed.

Function	Description
	Note The display deviates from the accompanying note display integrated in MVSC.
Display order file	The full order file is displayed.

The following figure shows an exemplary order overview:

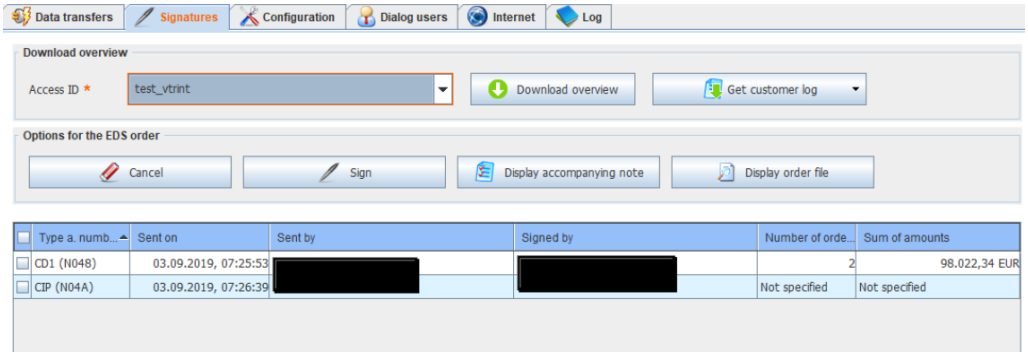


Figure 5.4. Order overview

5.2.4. Information on order files

Analysing file contents

Via the button "Analyse contents" the files listed in the table are analysed with regard to common order formats.

The following figure shows an exemplary file content:

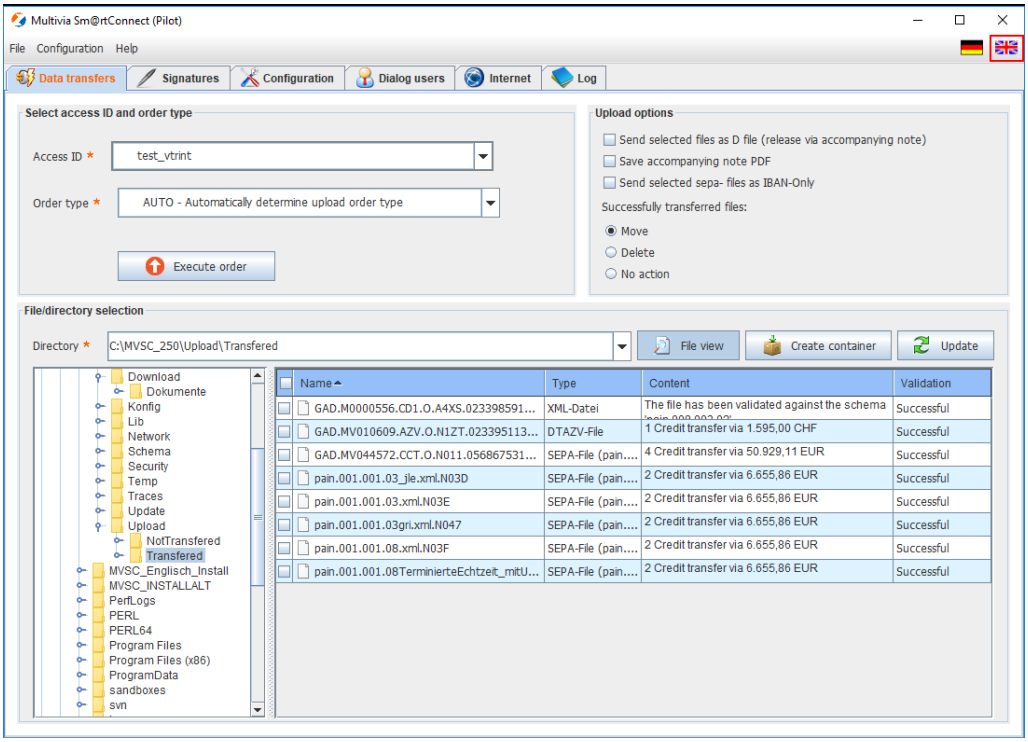


Figure 5.5. File content

The column "Type" displays the format of the respective file. The column "Contents" provides information on the order data in the files. The column "Validation" indicates whether the format of the order data is correct.

Note
The checks in this view are identical to the checks performed during the data transfer.

Displaying order data

By double-clicking on any entry in the table you can view the contents of the respective order file.

The following figure shows an exemplary payment file:

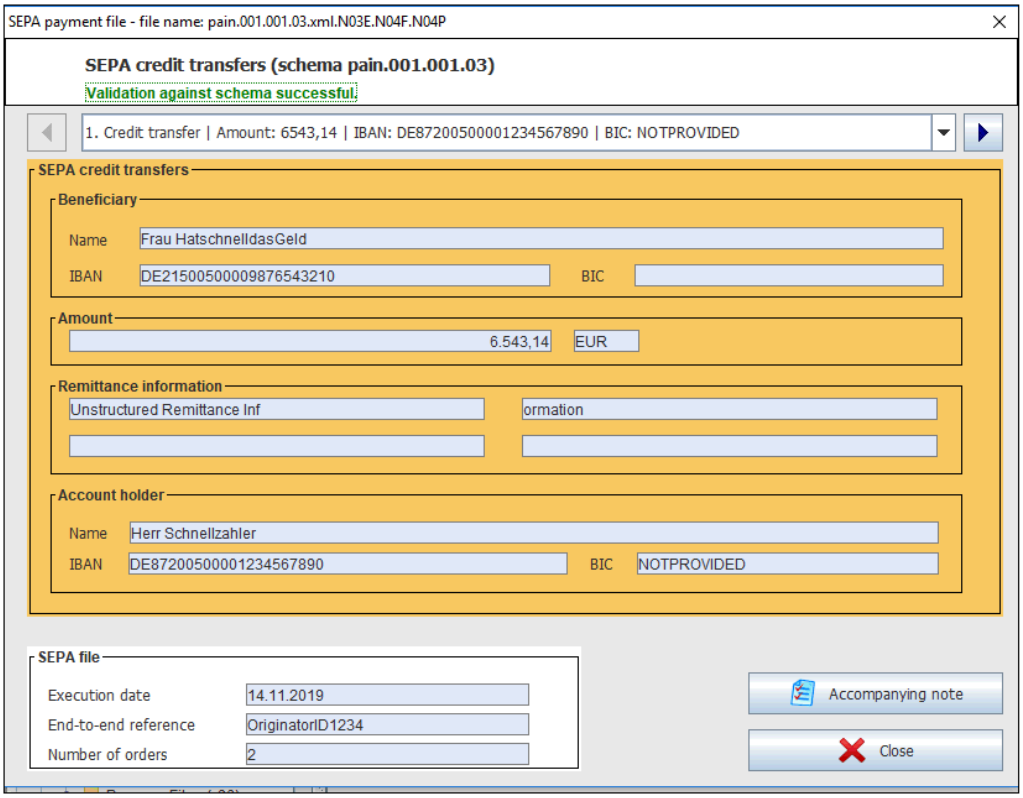


Figure 5.6. Order data display

The individual payment orders can be selected via the selection list located in the upper area of the mask. By means of the button "Accompanying note", a so-called " Accompanying note data carrier" can be generated. The result of the file validation is displayed below the title in green (validation successful) or red (validation failed).

5.3. Check options

Protocols in EBICS

Each action (upload/download of data) is written into a protocol at the EBICS bank server system. The entries and results in the protocol provide information on the processing status of the respective action or the submitted file.

The protocol can be downloaded and viewed via the order types "PTK" or "HAC". The contents of both protocols are identical; the difference lies only in how the information is processed.

PTK protocol

In the PTK protocol, the individual processing steps of uploaded orders are displayed **chronologically**. The information is provided in blocks. Each block contains information on one processing step. Additional actions may have been performed in between the respective processing steps; such actions are also included in the protocol. This means the protocol may include entries related to other actions between the upload information

and the order processing information. Based on the 4-digit order number (e.g. "N001"), you can see which PTK entry refers to which order.

The following diagram shows an example of the PTK protocol:

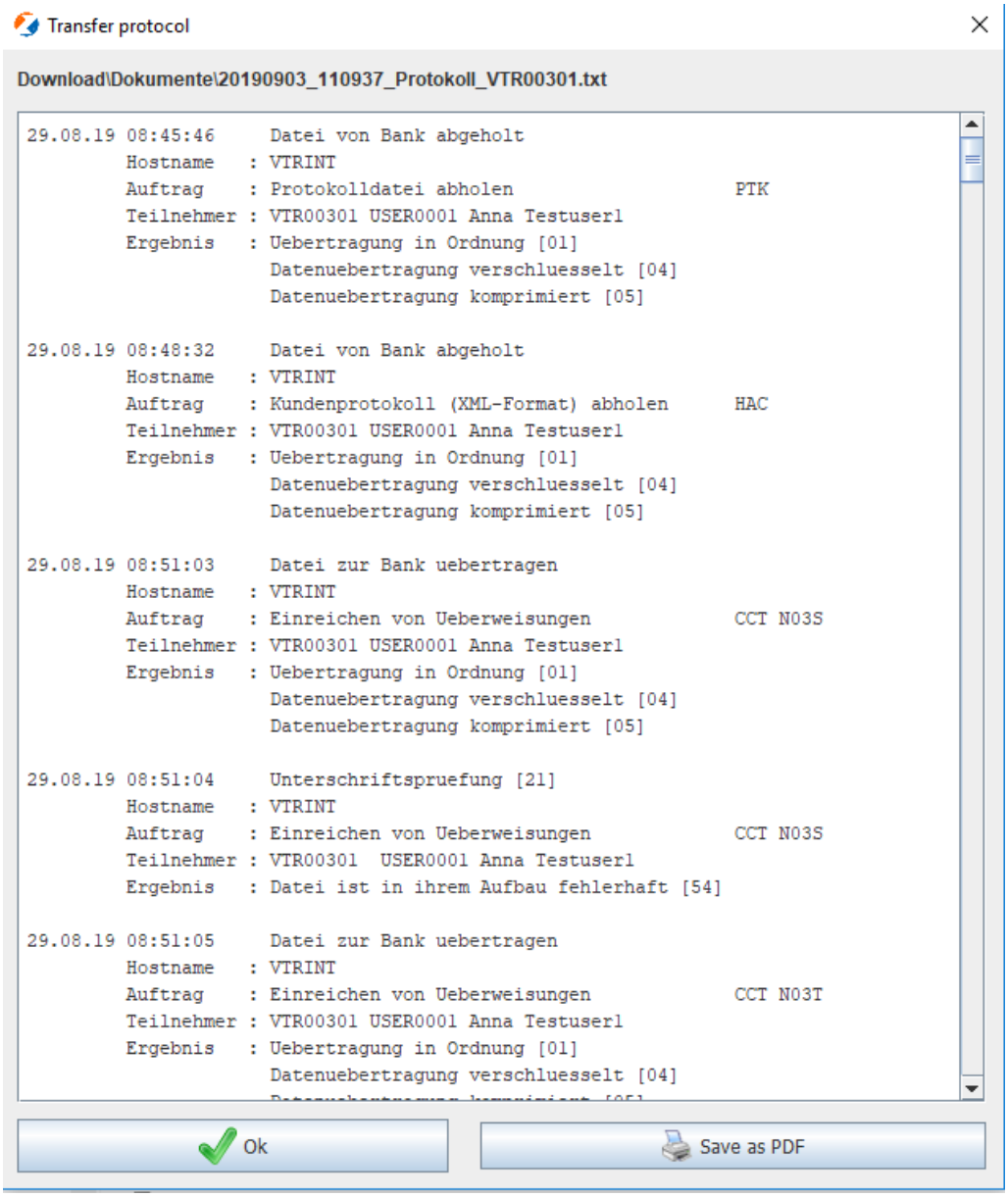


Figure 5.7. PTK transfer protocol

HAC protocol

The HAC protocol was introduced with the EBICS version 2.5. It provided the same information as the PTK protocol but in XML format. The advantage of HAC is that the return messages it contains can be evaluated automatically. The display in MVSC additionally offers you the option to filter using various criteria.

Tip
If you sort the entries by order type and order number, you can find entries related to a specific uploaded file faster.

The following figure shows the HAC customer protocol:

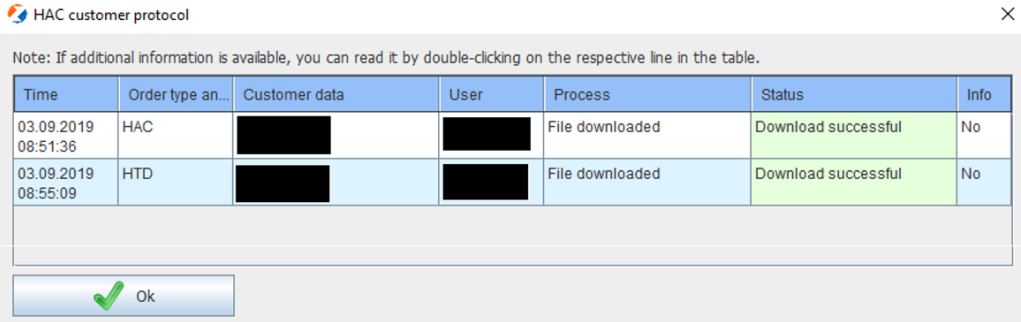


Figure 5.8. HAC customer protocol

The accompanying note known from PTK can be displayed via double-click if the column "Info" contains the value "Yes".

If the column "Process" contains the text "Order processing completed", this is the last protocol entry for the respective order. This entry provides information on the result of the processing.

Tip
The entries from the HAC protocol are also displayed in the order history. You can find more information on this in the following section "Order History".

Order History

The order history lists the entries of the HAC protocol for system-relevant order types such as "AZV", "CCT" and "CDS". Each line of the order history refers to an order for the respective customer ID and access ID.

The following figure shows an example of the "Order History":

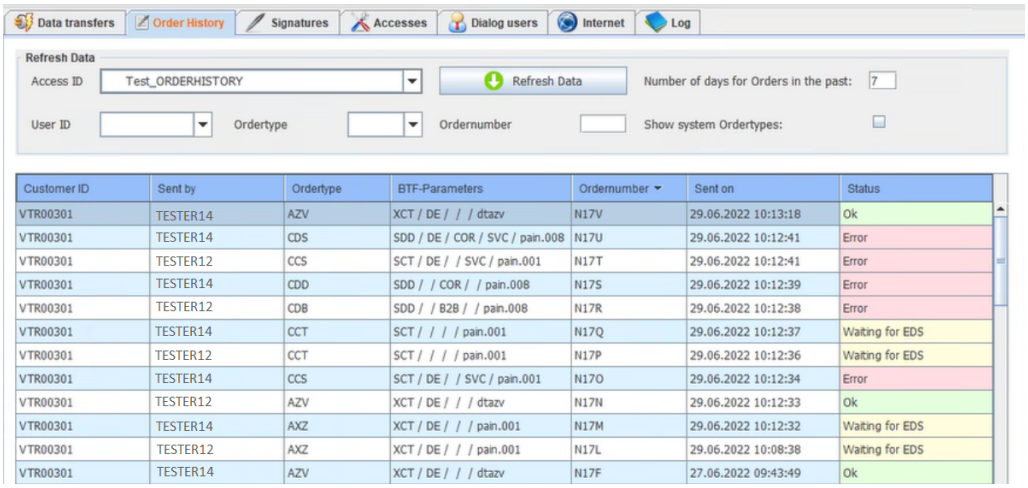


Figure 5.9. Order History

Press the "Refresh Data" button to fill or update the order history table based on the HAC log.

Use the field "Number of days for orders in the past" to define the time frame for which the order history is displayed. A specification between 1 and 90 days is possible.

Then you can double click on a line to retrieve extra information for the respective order.

The following illustration shows an example of the extra information for order:

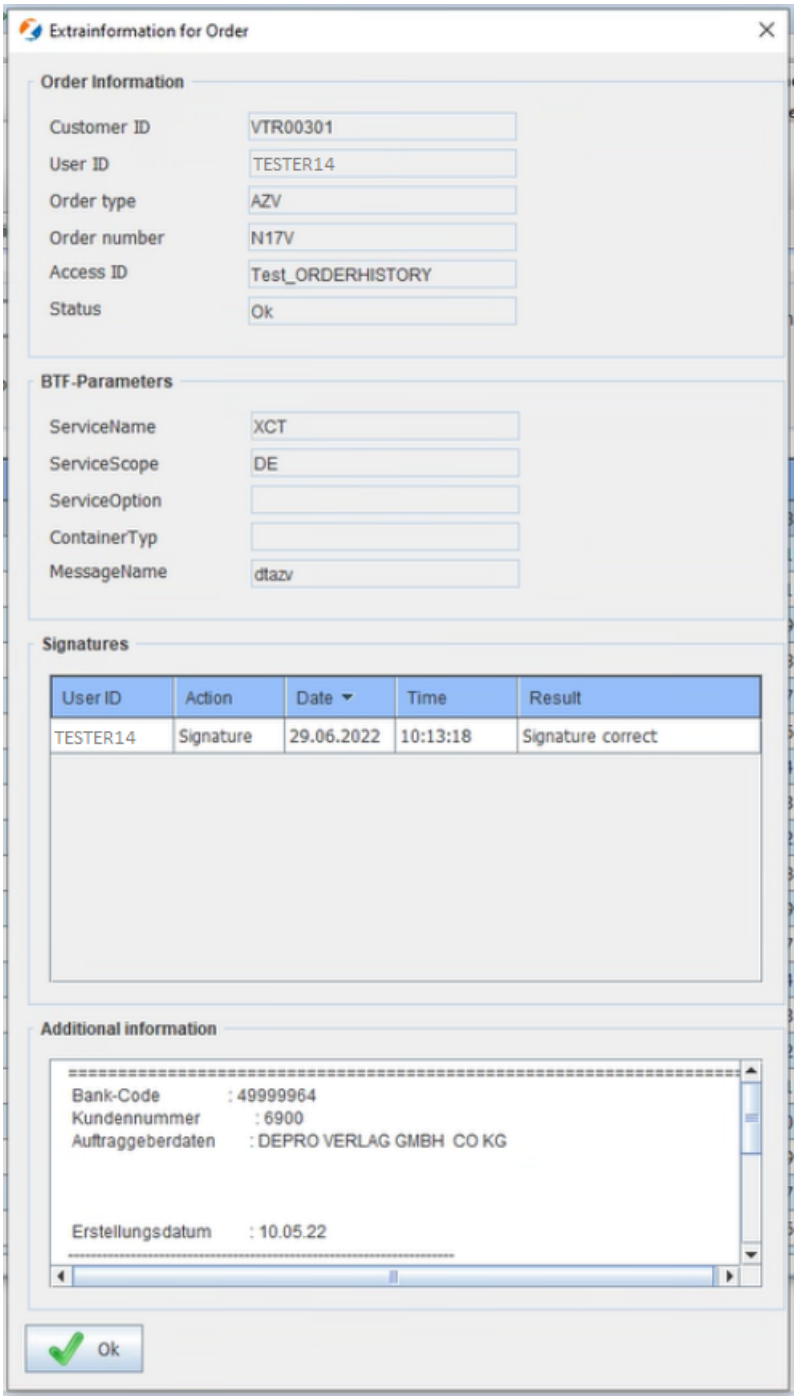


Figure 5.10. Order History - Extrainformation for Order

In the options, you set the time after which the order history is deleted. The options can be called up in the menu under the menu item "Configuration", "Options".

The following illustration shows how to call up the options:

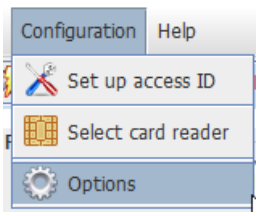


Figure 5.11. Order History - call up the options

The following figure shows the options with the indication "Days until the Order History gets deleted":

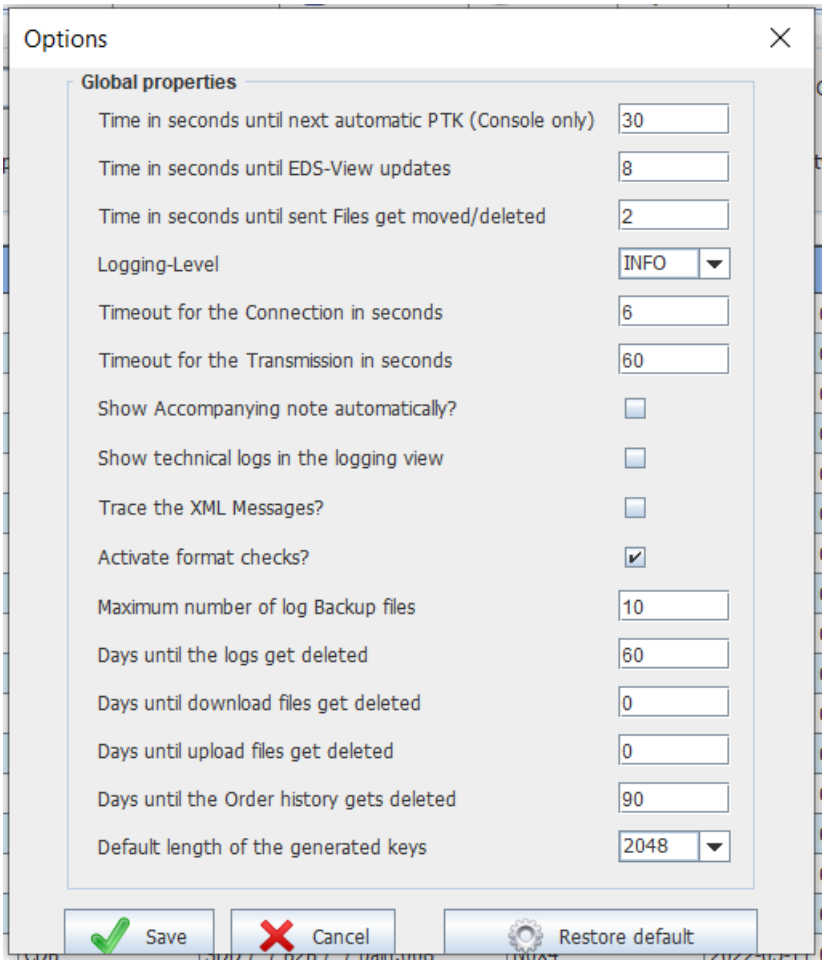


Figure 5.12. Order History - Options

This entry "Days until the Order History gets deleted" is to be selected depending on the order volume and the frequency of use. Older entries are then no longer displayed in the order history after they have been updated via the "Refresh data" button. They can be added back to the overview (depending on the field "Number of days for Orders in the past") by changing the entry and updating the overview again.

Tip
However, please note that orders older than 90 days that were once removed from the table cannot be added to the table again.

Specifying 0 days means that all existing orders from the order history are displayed. The default is 90 days.

Orders that are removed from the order history are automatically exported to a CSV file. This file is saved in the download directory. For more information on specifying the download directory in the default settings, see chapter "Default settings".

5.4. Use in console mode

Requirements

After all connection data (EBICS and Internet) has been entered and stored correctly in MVSC, the application can be used in console mode by means of a simple call. However, a prerequisite is that the passwords for the security medium and, if applicable, the proxy authentication are stored in the application.

Note
The console mode can only be used with access IDs for which a security file has been selected as security medium.

Preparations

Before using the console mode, some settings should be checked:

- Is the access ID fully initialised already?
- Is the configured security medium a security file?
- Is the password for the security file stored at the access ID?
- Are the settings in the dialog "Default settings" correct?
- Is the Internet connection established via a proxy server and is possibly required authentication information stored in the tab "Internet"?

If these prerequisites are fulfilled, a console call can be performed.

Note
To get to the dialog "Default settings", go to the tab "Accesses". Select the access ID that shall be used for the console call and click on the button "Default settings".

Call from the console

Once all aforementioned conditions are fulfilled, open your console (Start->Execute->cmd) and go to the MVSC installation directory:

```
cd C:\installation\directory\MVSC\
```

Start MVSC with at least one call parameter (name of the access ID that shall be used to transfer and/or download data).

The following call variants are available:

1. **Variant A: only the access ID is entered, all other parameters are determined from the configuration.**

   ```
   LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID"
   ```

   The order type stored for this access ID under "Accesses ->Default settings" is executed.

   If an upload order type is stored, all files in the specified upload directory can be transferred that match the file filter configured for the order type.

   If a download order type is stored, the received data is saved to the specified download directory (or documents directory).

2. **Variant B: the access ID and the order type are entered.**

   ```
   LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type"
   ```

   The entered order type is executed. For this variant, as well, the file filter configured for the order type is applied to the specified upload directory in case of an upload (see variant A).

3. **Variant C: the access ID, the order type and the upload and/or download directory are entered. For upload order types, an action parameter and two directories can be specified additionally.**

   ```
   LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type"
   "path/to/directory"
   ```

   The entered order type is executed.

   For upload order types, the third parameter is interpreted as the upload directory. That means that the file filter valid for the order type is applied to this directory path.

If a download order type is entered, the received data is saved to the entered target directory.

Note
For the order types "HAC" and "PTK", the entered path is ignored; instead, the documents directory is used as storage location.

For upload order types, an action parameter and two directories can be specified additionally.

In this case the call is as follows:

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type"
"path/to/directory" "action parameter" "path/to/move"
"path/for/erroneous/transfers"
```

The optional action parameter can assume the following values:

Action parameter	Meaning
"-verschieben"	Successfully transferred files are moved to the directory "path/to/move". If this directory is not specified in the call, the configuration from the Default settings is used.
"-loeschen"	Successfully transferred files are deleted. If a move path is entered, it is ignored.
"-keineAktion"	The transferred files are neither moved nor deleted. If a move path is entered, it is ignored.

If an error occurs during the file transfer, the file to be transferred is moved to the directory "path/for/erroneous/transfers".

If this directory is not specified in the call, the configuration from the Default settings is used.

4. **Variant D: the access ID, the order type and the upload file are entered. An action parameter and two directories can be specified additionally. This variant is only relevant for upload order types.**
   Upload order types:

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type"
"path/to/file"
```

The entered file is transferred in consideration of the specified access ID and order type. The file is neither moved nor deleted via this call.

However, an action parameter and two directories can be specified additionally.

In this case the call is as follows:

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type"
"path/to/file" "action parameter" "path/to/move" "path/for/erroneous/transfers"
```

The optional action parameter can assume the following values:

Action parameter	Meaning
"-verschieben"	Successfully transferred files are moved to the directory "path/to/move". If this directory is not specified in the call, the configuration from the Default settings is used.
"-loeschen"	Successfully transferred files are deleted.
"-keineAktion"	The transferred files are neither moved nor deleted. If a move path is entered, it is ignored.

If an error occurs during the file transfer, the file to be transferred is moved to the directory "path/for/erroneous/transfers".

If this directory is not specified in the call, the configuration from the Default settings is used.

Download order types:

For download order types (e.g. "AUTD"), the call can be made in two different ways.

The call

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type
download"
```

is to be used by default.

Alternatively, especially if account transactions are to be retrieved for a specific date or for a specific period, the following call can be used:

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type
download" "path to download directory" "action parameter"
```

The two parameters "path to download directory" and "action parameter" are therefore optional.

The following action parameters can be used as an alternative:

Aktionsparameter	Bedeutung
"-Datum="	For account transactions, enter the date "from", "to" as follows: "dd.mm.yyyy,dd.mm.yyyy" Example: "-Datum=30.09.2024,15.10.2024" In this example, the account transactions from 30 September up to and including 15 October are retrieved.
"-Zeitraum=" (This means time period)	For account transactions, the two numerical values "Number of days from" and "Number of days to" must be entered here. The first number corresponds to the starting point (number of days in the past), the second number corresponds to the end point (number of days in the past). Example: "-Zeitraum=2,1": This information includes the days "the day before yesterday" and "yesterday". Example: "-Zeitraum=2,0": This information includes the days "the day before yesterday" and "yesterday" and today. Example: "-Zeitraum=0,0": This information only includes today's date.

It is recommended to specify the period of time.

Order types 'AUTO' and 'AUTD'

With the upload order type 'AUTO' all upload files from the respective upload directory with the corresponding order type are transferred to the bank server.

Note
In contrast, if the order type 'CCT' is specified, for example, only files with the order type 'CCT' are transferred to the bank server.

When using the download order type 'AUTD' all download order types, which are valid for the specified access ID, are processed.

(e.g. 'LibJAV\bin\java MVSC.jar ZugangsIdTest1 AUTD c:\MVSC\Download\ ')

A single call of CAMT or MT940 account turnover data is therefor not necessary.

Tip
If you do not want a PTK/HAC file to be created during the download in console mode, you can set the corresponding entry in 'Db\mvsc.properties' to 'ON' ('hac.before.download=ON'). By default this value is 'OFF'.

Workflow

No other specifications are possible or required after this. MVSC recognises the entered access ID and searches the specified upload directory for files that match the file filter configured for the order type. These files are transferred to the EBICS bank server system one after the other.

Caution
If the application is called periodically (e.g. every 30 minutes), you must make sure that already transferred files are not transferred again within the following application call. It is therefore recommended to configure MVSC to move or delete such files. Otherwise the caller himself must ensure that the files are not transferred multiple times.

Returns

Once the transfers are completed, MVSC returns a value that provides information on whether the action was successful. Details on the individual return values can be found in the chapter "Return values in console mode".

Tip
The information output on the console during the transfer can be rerouted into a file via the operating system. To this end, that target file must be specified behind the respective MVSC call either with one ">" character if you want to overwrite it or with two ">" characters if you want to continue it.

Example: "Call variant >> MVSC_call.log"

5.5. Automated use via batch file

Integration into complex processing workflows

If MVSC as transfer component shall be integrated into a complex overall workflow, this can, for example, be realised via a so-called batch file. The application call is then executed from a self-provided framework application that runs through a more or less complex process automatically.

The following table describes a relatively simple example:

Action	Process/result
Call from an accounting software	Payment files with the file extension ".xml" have been created in a defined outgoing directory.
MVSC call (matches variant B of the call variants)	**Prerequisite according to variant B:** The access ID and the order type are provided to MVSC as call parameters. The outgoing directory is preset as upload directory at the access ID. For the specified order type, the file extension ".xml" is configured in the file filter. **Result of the call:** All files in the outgoing directory that have the file extension ".xml" are transferred to the EBICS bank server with the specified access ID.
Query for MVSC return value	The value returned by MVSC must be evaluated. Based on this value it can be determined how to proceed in the overall process.

Example for a batch file

A batch file has already been delivered with MVSC. It is called "beispiel_batch.cmd" and can be found in the MVSC installation directory.

The file is only intended as an example and cannot be used in its delivery state. In the example, MVSC is called once every 10 minutes starting from the time of its first call until 11 pm. Afterwards, certain return values of the application are evaluated and output.

The following line must be adjusted so that the file can be executed (line 46 in the file):

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" <$MY_ACCESSID> <$MY_ORDERTYPE>
<$MY_UPLOADDIRECTORY>
```

Here the access ID, the order type and the upload directory must be adjusted. Of course, a different call variant may also be used.

5.6. Default settings

General information

The dialog "Default settings" is mostly used to configure the console mode. It indicates, for example, the directories in which the application searches for files matching a certain pattern upon the respective call.

Additionally, the dialog also contains some settings that affect the use in the GUI.

Figure of the dialog

The following figure shows the mask "Preset default settings":

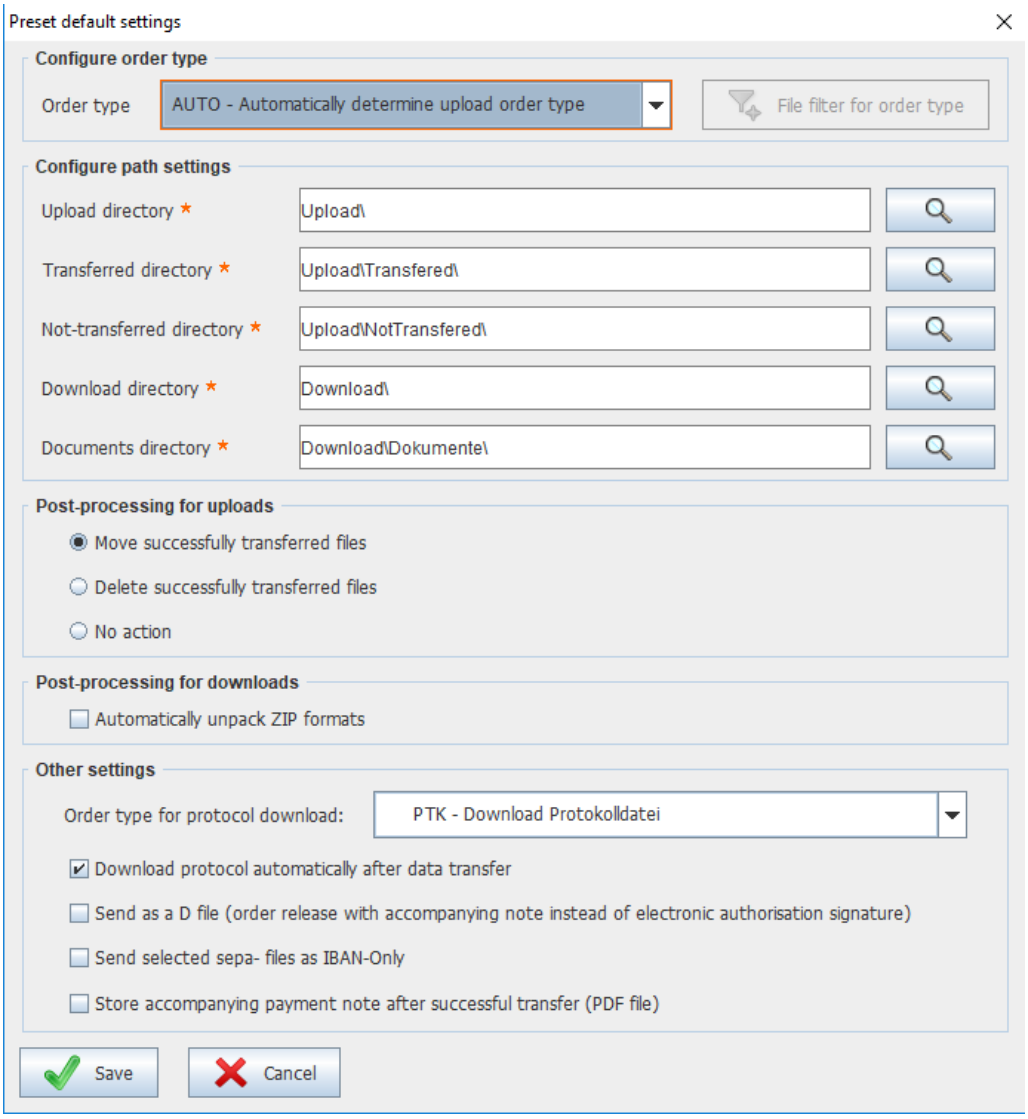


Figure 5.13. Preset default settings

Group "Configure order type"

The Order type in the selection field is executed whenever the dialog user calls the application in console mode with the respective access ID as specified parameter. If an

upload order type is selected, the button "File filter order type" is activated. A click on this button opens the so-called "file filter editor". In the editor, you can specify file extensions (e.g. "txt") that shall be transferred with the currently selected order type.

Group "Configure path settings"

Here you configure the storage directories for the selected access ID.

- **Upload directory:**
  When the application is called with the parameter "access ID", this directory is searched for files that match the file filter configured for the respective order type. All files found matching that filter are automatically transferred with this order type.

  In the mask "Data transfers", this directory is preset if an upload order type was selected.
- **Transfered directory**
  Files that were transferred by MVSC are moved to this directory so that they cannot be transferred more than once upon the next application call.
- **NotTransfered directory**
  Files that could not be transferred successfully in console mode are moved to this directory. In GUI mode, files that could not be transferred successfully remain in their original storage location. The directory is thus only relevant for console mode.
- **Download directory:**
  All files that were received via a download order type are stored in this directory. This excludes information from the order types "PTK" (download protocol file) and the initialisation process (INI letters).

  In the mask "Data transfers", the download directory is preset if a download order type was selected.
- **Documents directory**
  Protocol files downloaded from the bank server (order type "PTK") are stored in this directory. Additionally, it serves as storage location for INI letters.

Group "Post-processing for uploads"

After a file has been transferred to the EBICS bank server, there are three options for how to proceed with that file:

- "Move successfully transferred files": the file is moved to the specified Transfered directory. (standard)
- "Delete successfully transferred files": the transferred file is deleted.
- "No action": no post-processing takes place and the transferred file remains unchanged.

The selected post-processing procedure only takes place after the file has been successfully transferred to the EBICS bank server.

Caution
If you are working in console mode, we recommend moving or deleting the files as they could be transferred again upon the next application call. If "No action" is selected, the calling process must ensure that the files cannot be transferred more than once.

Group "Post-processing for downloads"

If the option "Automatically unpack ZIP formats" is activated, MVSC automatically unpacks download files in ZIP format (e.g. CAMT files).

The subdirectory generated for the unpacked files always has the same name as the download file without its file extension.

Example:

If a file was stored under the name "C53_20131216_135056_CUSTOMERID_USERID.C53", it is unpacked to the directory "C53_20131216_135056_CUSTOMERID_USERID".

Group "Other settings"

These settings apply for both the console mode and data transfers via the GUI:

- **Order type for the protocol download:**
  As of EBICS version 2.5 there are two order types for downloading the customer protocol: "PTK" and "HAC". While the order type "PTK" delivers the protocol in text format, the order type "HAC" provides the same information in a machine-readable XML format.

  This field indicates which order type shall be used for the download and thus which format the customer protocol shall have.

  Note
  If the order type "HAC" is not assigned to the respective access ID, the selection list is deactivated. The order type "PTK" is used in this case.

- **Send as a D file:**
  This option can be used to transfer files to the bank server without electronic signatures (ES). The orders transferred this way must be released by means of a signed accompanying note which has to be provided to the bank on paper.
- **Send selected sepa-files as IBAN-Only:**
  If this checkbox is activated, all contained BICs will be removed. The original file is then located in the directory "Upload/Original". However, the transferred file is then in the directory "Upload/Transfered".

  This option can be used to transfer files to the bank server without electronic signatures (ES). The orders transferred this way must be released by means of a signed accompanying note which has to be provided to the bank on paper.
- **Protocol after data transfer:**
  In console mode, the customer protocol is automatically downloaded after the transfer of the order files. This option does not affect the operation in GUI mode.
- **Store accompanying payment note after successful transfer:**
  If this checkbox is activated, an accompanying note in PDF format is stored for each successfully transferred payment file.

5.7. SDC functions

General information

The function described in the following section are mainly relevant for service data centres.

Multivia Sm@rtConnect supports the creation of SEPA XML containers including hash values. It also enables the creation of accompanying notes according to the SDC guidelines.

The SDC functions can be found in a separate dialog that is opened via the button "SDC functions" in the tab "Accesses". The settings you configure always apply to the selected access ID.

Figure of the dialog

The following figure shows the mask "SDC settings":

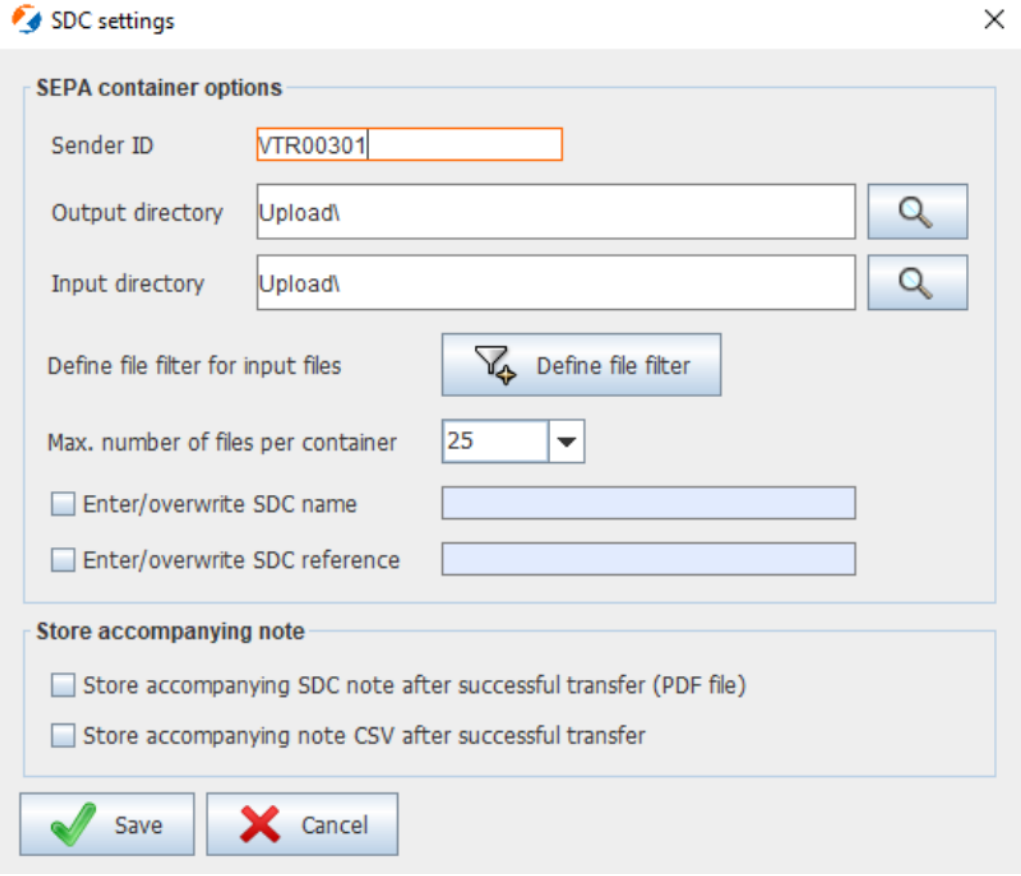


Figure 5.14. SDC settings

Group "SEPA container options"

The settings in this group refer to the call for container creation in console mode. They do not apply to the container creation in GUI mode.

- **Sender ID:**
  The sender ID is the identification of the file sender on the respective target system. If the file is transferred via EBICS, the EBICS customer ID must be entered here. It is used as a standard when you create an access ID.

  Note
  Aside from the sender ID, MVSC fills the field "Identification Type" with the value "EBIC". This value indicates that the specified sender ID originates from the EBICS procedure.

- **Output directory:**
  In the directory indicated here, the XML container generated during a console call are stored.
- **Input directory:**
  This directory is searched for files that are compiled into an XML container. The input files must meet certain conditions so they can be compiled into a container. More information on this can be found in the section "Requirements for container creation". In console mode, only files matching the configured file filter are considered.
- **Define file filter for input files:**
  The file extensions configured here define which file types shall be considered for the container creation. The file filter works according to the same principle as the dialog "Default settings".
- **Max. number of files per container:**

The value specified here defines the number of files after which MVSC shall create a new XML container.

Note
Please note that even in case of few input files several containers may be created. This is due to the fact that certain business-related and technical rules must be observed for the container creation, which requires several containers to be created.

An example of this is the separation of credit transfers and direct debits ("COR1", "CORE", "B2B").

- **Enter/overwrite SDC name:**
  The name of the party submitting the file (SDC name) can be specified in the SEPA source files to be compiled into a container. If the name of the submitting party was not specified or was specified incorrectly by the system creating the file, the name can be added or overwritten later with MVSC. If this option is activated, MVSC enters the SDC name specified in this field into **all** SEPA files to be compiled into the container.

  To this end, the following XML structure (in bold) is added or adapted in each source file:

  <GrpHdr>**<InitgPty><Nm>SDC NAME</Nm></InitgPty>**</GrpHdr>
- **Enter/overwrite SDC reference:**
  Just as the SDC name, the reference of the file submitter (SDC reference) can be specified in the SEPA source files. If the corresponding checkbox is activated, the specified SDC reference is entered in **all** SEPA source files that are added to the container.

  To this end, the following XML structure (in bold) is added or adapted in each source file:

  <GrpHdr>**<InitgPty><Id><OrgId><Othr><Id>SDC REFERENCE</Id></Othr></OrgId></Id></InitgPty>**</GrpHdr>

Group "Store accompanying note"

The settings for the accompanying note made here apply for both the GUI and the console mode. The accompanying notes are only generated if the data was successfully transferred.

- **Store accompanying SDC note after successful transfer (PDF file):**
  If this checkbox is activated, an accompanying SDC note is stored for the transferred file. In case of DTAUS formats, the accompanying SDC note additionally contains the reference data specified in the "DTAUS options". For SEPA files, the accompanying note is generated with the hash value contained in the respective file. It also contains the information from the GroupHeader (SDC name/SDC reference).
- **Store accompanying CSV note after successful transfer:**
  If this option is activated, the accompanying note information is stored in the format CSV (Comma Separated Values). For each logical unit in the order file, one line is generated in the CSV file. The semicolon is used as delimiter between the individual values. The information in each line equals the information in the PDF accompanying notes.

5.8. Container creation

General information

With MVSC, individual SEPA XML files that meet certain conditions can be compiled into an XML container format. During the container creation, the individual input files are embedded into the container format in accordance with the SDC guidelines. To this end, the input files are canonicalised first. Then the so-called hash value (SHA-256) of the

documents is generated. The hash value serves as check mechanism for the individual documents and is therefore not displayed in the accompanying note.

Requirements for container creation

The individual SEPA files in pain format must fulfil the following requirements so they can be compiled into an XML container:

- The input files must have one of the following formats: credit transfers: pain001, direct debits: pain008
- The files must be structured correctly according to the underlying pain format (check against the XSD schema).
- Only one element of the type "Payment Information (<PmtInf>)" per file is permitted. This element indicates a payment that can consist of various transactions.

If the created container is to be submitted via SDC procedure, it is additionally recommended to specify the "name of the submitting SDC" and the "reference of the submitting SDC" in the so-called "Group-Header".

Note
MVSC does not check the values in these fields; thus containers can also be created without them. It is, however, possible to add or change this data later on. A description on this can be found in the section Group "SEPA container options".

Container creation in the GUI

The container creation can be found in the GUI and opened via the button "Create container" in the tab "Data transfers". The currently selected directory is applied as first source directory in the dialog. In the upper mask area you can select other source directories, as well. The left-hand area lists the content (files) of the selected source directory.

The entries listed here can be selected via their respective checkboxes and added to the container via the button with an arrow to the right.

Once all required files have been selected, you can start the container creation via the button "Create container". In the subsequent dialog, you must enter the sender ID and the output directory. Optionally, you can enter the "SDC name" and the "SDC reference".

Caution
When creating SEPA XML containers, the input files are separated by their pain format and by the direct debit procedure (core direct debit or B2B direct debit). Files of the same type are compiled into one container until they reach the maximum number per container specified in the "SEPA container options".

Figures of the dialogs

The following figure shows the mask for file selection:

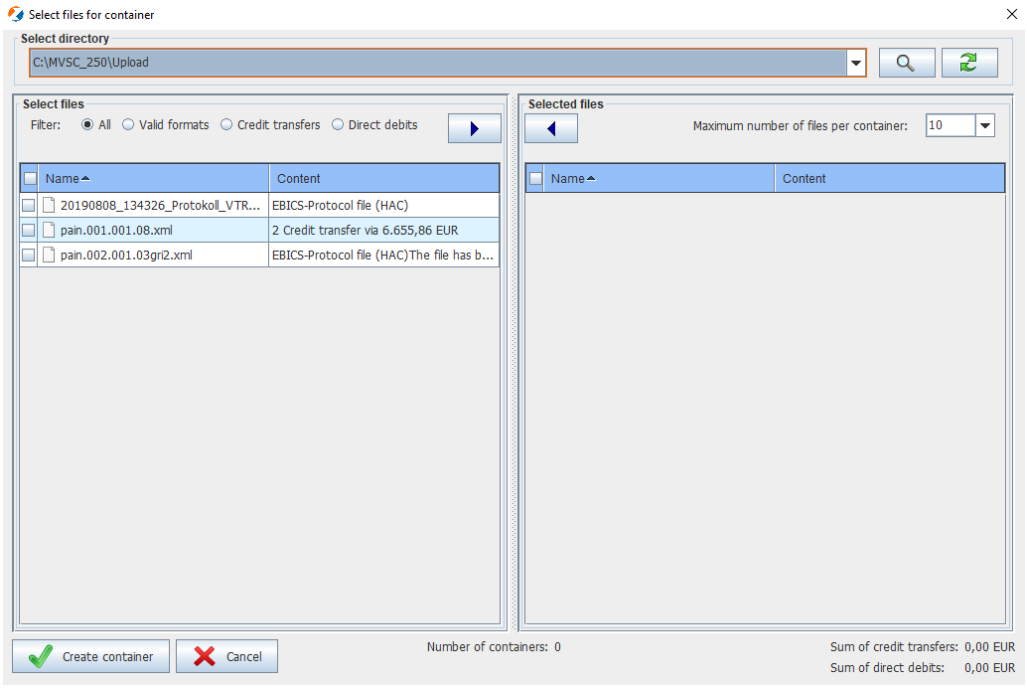


Figure 5.15. Selection of the files for a container

The following figure shows the mask for container creation:

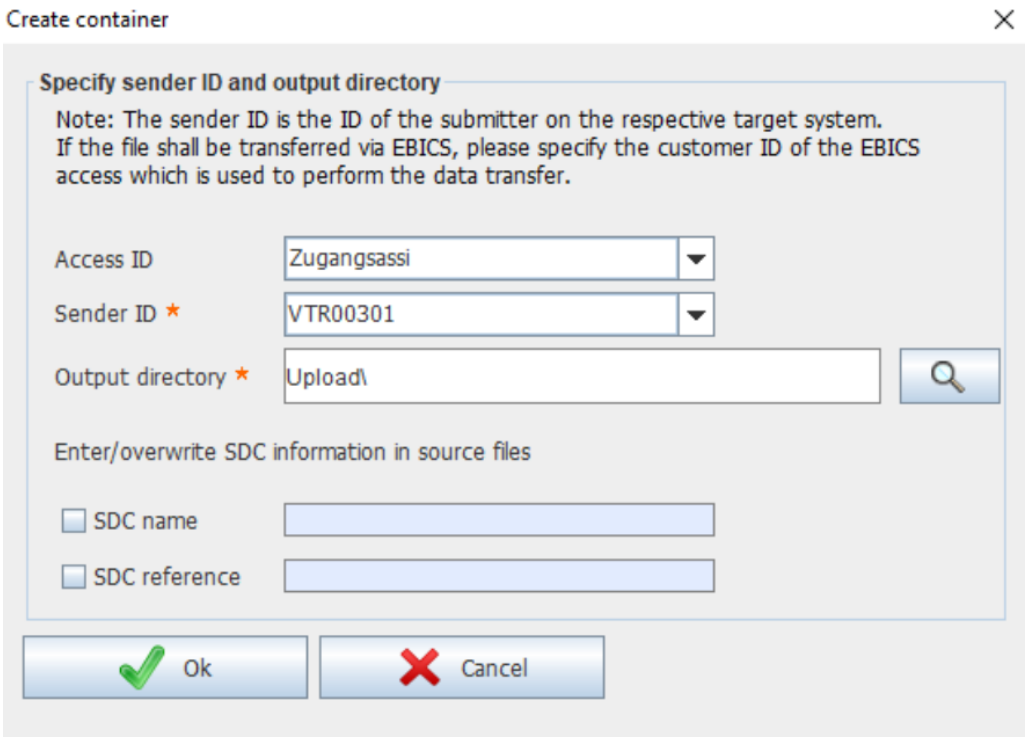


Figure 5.16. Container creation

You must enter the sender ID and the output directory here.

Structure of the file names

The created container files are stored in the specified output directory. The file names have the following structure:

<Timestamp of the creation>_CONTAINERCONTENT_SENDERID.SDCORDERTYPE

Example for credit transfers:

20131130_162449_CONTAINER_GUTSCHRIFTEN_MYID1.CCS

Example for direct debits:

20131130_162449_CONTAINER_LASTSCHRIFTENCORE_MYID1.CDS

20131130_162449_CONTAINER_LASTSCHRIFTENB2B_MYID1.C2S

20131130_162449_CONTAINER_LASTSCHRIFTENCOR1_MYID1.C1S

If multiple files are created at the same time, one more counter is added as follows:

20131130_162449_CONTAINER_LASTSCHRIFTENCORE_MYID1_**1**.CDS

Container creation in console mode

The container creation can also be called via command line. To this end, go to the MVSC installation directory as follows:

```
cd C:\installation\directory\MVSC\
```

Start the container creation with the desired parameters:

1. **Variant A: only the access ID and the identifier for the container creation are entered. All other parameters are determined from the configuration.**

   ```
   LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "CONTAINER"
   ```

   For this call variant, the settings stored in the dialog "SDC functions" are used. The configured file filter is applied to the specified input directory. The input files determined by the filter are validated and embedded into the container format. The created containers are stored in the specified output directory.

   Note

   The so-called "sender ID" and the parameter "Maximum number of files per container" are always determined from the configuration. These values can therefore not be specified via the command line. The file filter must also be configured in the "SDC functions" before the call.

2. **Variant B: the access ID, the container identifier and the input directory are entered.**

   ```
   LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "CONTAINER"
   "path/to/input/directory"
   ```

   This call differs from variant A only in one instance:

   The input directory stored in the configuration is ignored. Instead, the directory path entered in the call is used as input directory. All other parameters are determined from the configuration.

3. **Variant C: the access ID, the container identifier and the input directory as well as the output directory are entered.**

   ```
   LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "CONTAINER"
   "path/to/input/directory" "path/to/output/directory"
   ```

   As in variant B, the input directory is also entered here. In this call, however, the output directory is also specified. It is used instead of the output directory stored in the configuration.

Return values and error handling

MVSC checks whether files that are to be added to a container in generally suited for container formats. It is checked whether the conditions described under "Requirements" are fulfilled.

If this is not the case, a corresponding message is displayed in the GUI. The file is not added to the container.

If this situation occurs in console mode, however, the container creation is aborted: no container is created and the application returns a corresponding return value.

The following table lists the possible return values of the console mode:

Return value	Meaning
-1	The EBICS access data is incomplete or erroneous.
-4	No input files were found (according to the file filter and input directory).
-6	Parameter error: the call parameters are incorrect.
-8	At least one directory could not be found (validation of the input/output directories).
-9	Double call: the application is already running.
1	**The container creation was successful; at least one container has been created.**
31	The file format of at least on input file is unknown.
32	XML format invalid: one file does not match the SEPA formats suited for containers.
33	XML validation failed: one file contains formatting errors according to the XML schema.
34	XML validation: the file contains more than one <PmtInf> block (multiple logical file).
35	XML validation: the Service Level is invalid for containers.
36	XML-validation: the Local Instrument is invalid for containers.
37	XML creation: an error occurred during the canonicalisation of an input file.
38	XML creation: an error occurred during the hash value calculation.
39	XML creation: a technical error occurred during the container file creation.

Generally, the negative return values indicate configuration and/or call issues, whereas the values in the 30s can be attributed to errors in the input files (31 to 36) or errors during the creation of the container files.

Creating containers

By means of the different call variants in console mode, the container creation can be combined with the file transfer. To do that, the settings used by MVSC as a standard are already sufficient.

The following prerequisites must be fulfilled:

- The calls must be executed with a fully initialised access ID (including the download of the order types).
- The input directory and the file filter in the dialog "SDC functions" may have to be adapted to your system/your input files.
- The specified input directory must contain files that match the configured file filter (the standard value is "XML"). Information on further requirements for the container creation can be found in the section "Requirements for container creation".
- The upload directory of the access ID was specified as output directory for the containers (standard).

Go to the MVSC directory and execute one of the calls described in the section "Container creation in console mode".

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "CONTAINER"
```

If the call was successful (return value 1), at least one container file was created and stored in the output directory. The file name structure of the created container files can

be found in the section "Structure of the file names". The file names always end with the required EBICS order type.

Sending containers

The following call is used to send the container:

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" "order type"
```

If different container types are created out of your input files, you must initiate the data transfer with different order types to ensure that all files are transferred.

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" CCS
```

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" CDS
```

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" C1S
```

```
LibJAV\bin\java -Xms96m -Xmx256m -jar "MVSC.jar" "MY_ACCESS_ID" C2S
```

These calls initiate the transfer of all files from the upload directory that match the file filter configured for the specified order types.

As a standard, MVSC stores the ID of the order type in the individual file filters. The created container files thus match the file filters of the respective order type.

5.9. Verification of payee

Precondition

Since October 2025, specifications have been in place for SEPA credit transfers and SEPA instant credit transfers: It has to be be possible to compare the payee details with the payee data stored at the bank for the specified IBAN.

In order to be able to use this recipient verification (hereinafter also referred to as 'VoP' - **V**erification **o**f **P**ayee) in MVSC, the customer requires the following new opt-in order types:

Order types for submission: 'CIV' (Collective transfer of SEPA-Real-Time-Transfers with VoP), 'CTV' (SEPA-Transfer with VoP)

Order type for pick up: 'VPZ' (VoP status report)

You can view the order types by clicking on the 'Accesses' tab and then on the 'Download authorisations' button. This button is marked in red in the lower part of the screen in the following illustration:

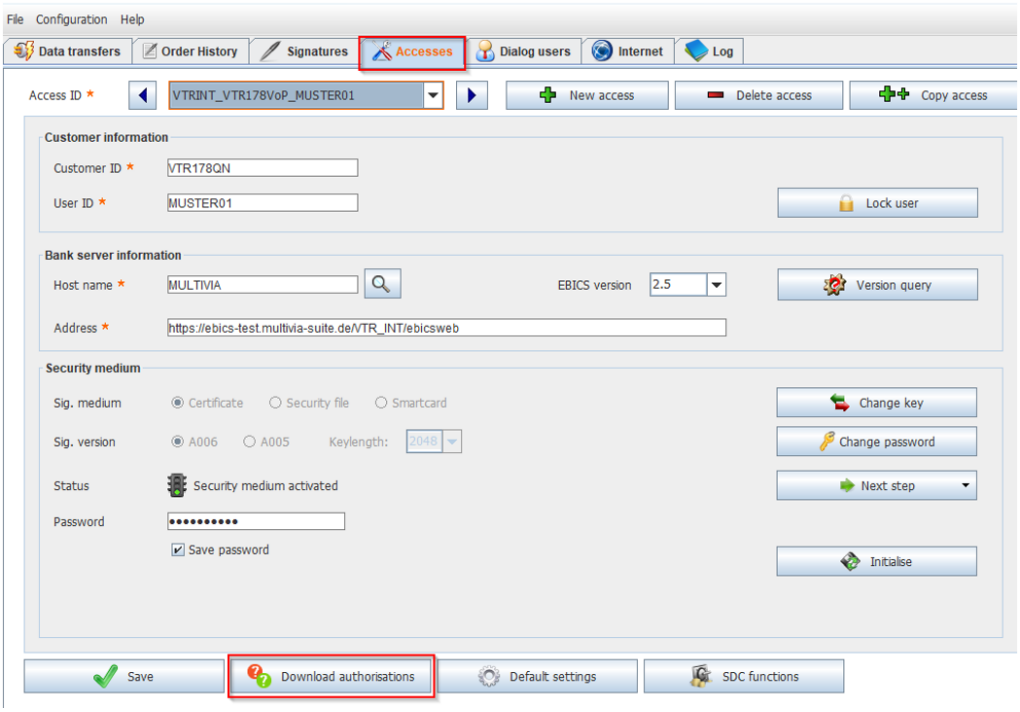


Figure 5.17. VoP: Download authorisations

After the order types have been retrieved, they are entered into the MVSC database. This is illustrated by order types "CIV" and "CTV" in the following figure:

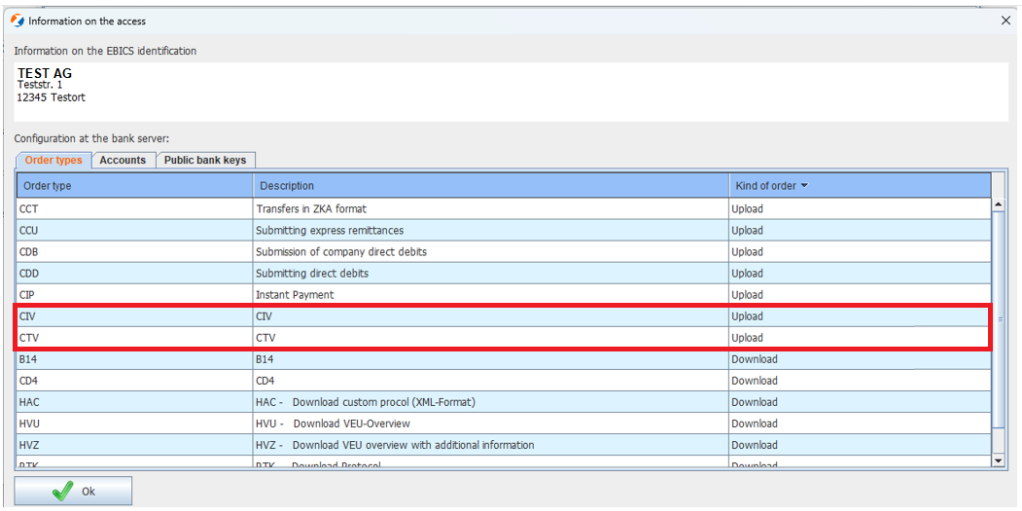


Figure 5.18. VoP: Result of the order type retrieval

Default settings

You can make specific default entries for recipient verification ('VoP'). To do this, select the 'Default settings' button at the bottom of the screen under the 'Accesses' tab.

This button is marked in red in the lower section of the following illustration:

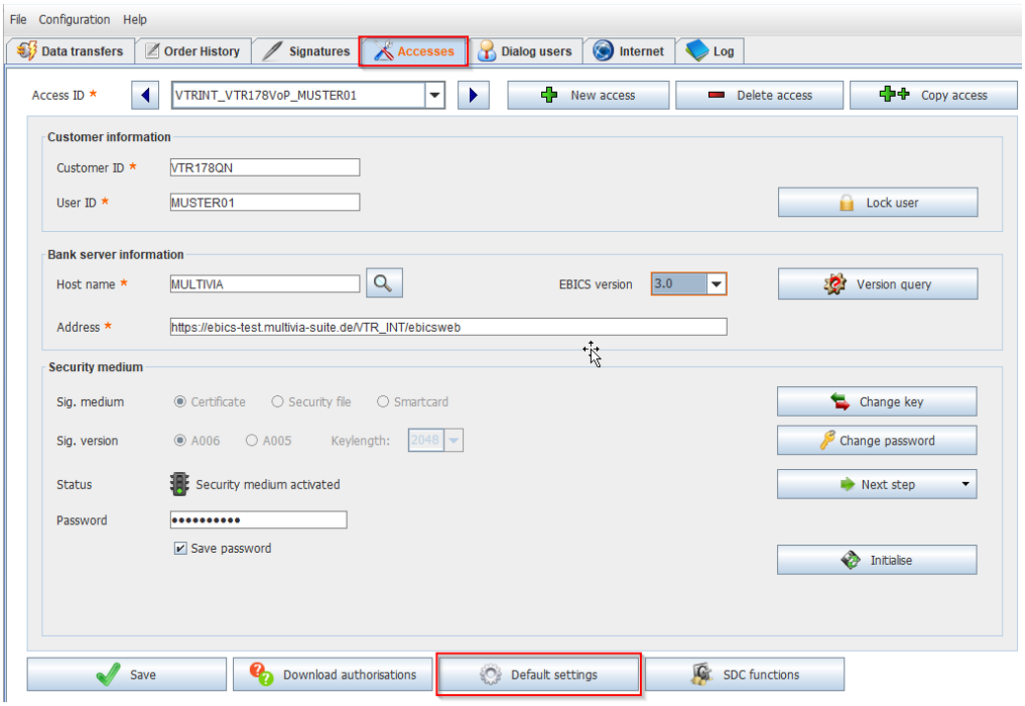


Figure 5.19. VoP: Call Default settings

You will then see the screen shown in the following illustration:

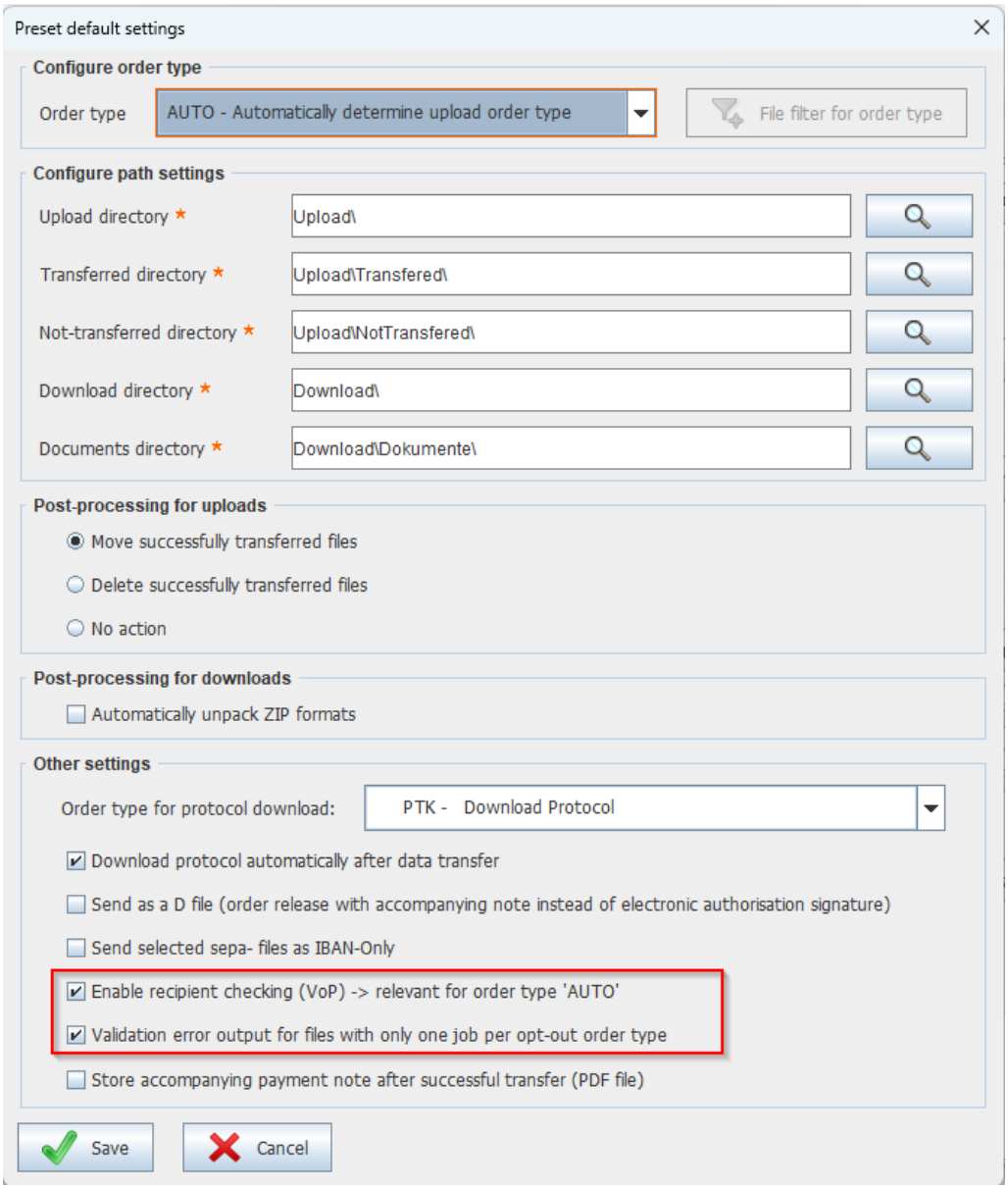


Figure 5.20. VoP: Make Default settings

Here you will find the two checkboxes marked in red at the bottom of the screen shown:

- 'Enable recipient checking (VoP) -> relevant for order type 'AUTO'

and

- 'Validation error output for files with only one job per opt-out order type' -> Validation error output for files with only one request per opt-out request type (i.e. without recipient verification)

You can enable or disable the recipient verification for the 'AUTO' order type using the check box **‘Enable recipient checking (VoP)’**.

When activated, this means that the opt-in order types (i.e. order types with recipient verification, e.g. ‘CTV’, “CIV”) are used for this access when selecting the order type ‘AUTO’ in data transfer.

The **checkbox ‘Validation error output for files with only one job per opt-out order type’** allows to control whether an error message should be generated if a data transfer with an opt-out order type (i.e. without recipient verification) is performed and the file to

be transferred contains only one order. Such an error message is only issued if the checkbox is activated. There is no automatic switch to an opt-in order type.

Data transfer with or without recipient verification in the graphical user interface (GUI)

First, check that the two checkboxes described in the section above entitled "Default settings"

- Enable recipient checking (VoP) -> relevant for order type 'AUTO"

and

- Validation error output for files with only one job per opt-out order type -> Validation error output for files with only one request per opt-out request type (i.e. without recipient verification)

are correctly maintained.

No further information is required for data transfer.

Data transfer with 'VoP' (opt-In request type, i.e. with recipient verification) using the order type 'AUTO':

Precondition: The access includes activating the ‘Enable recipient checking (VoP)’ checkbox in the default settings.

The order type ‘AUTO’ is selected for data transfer.

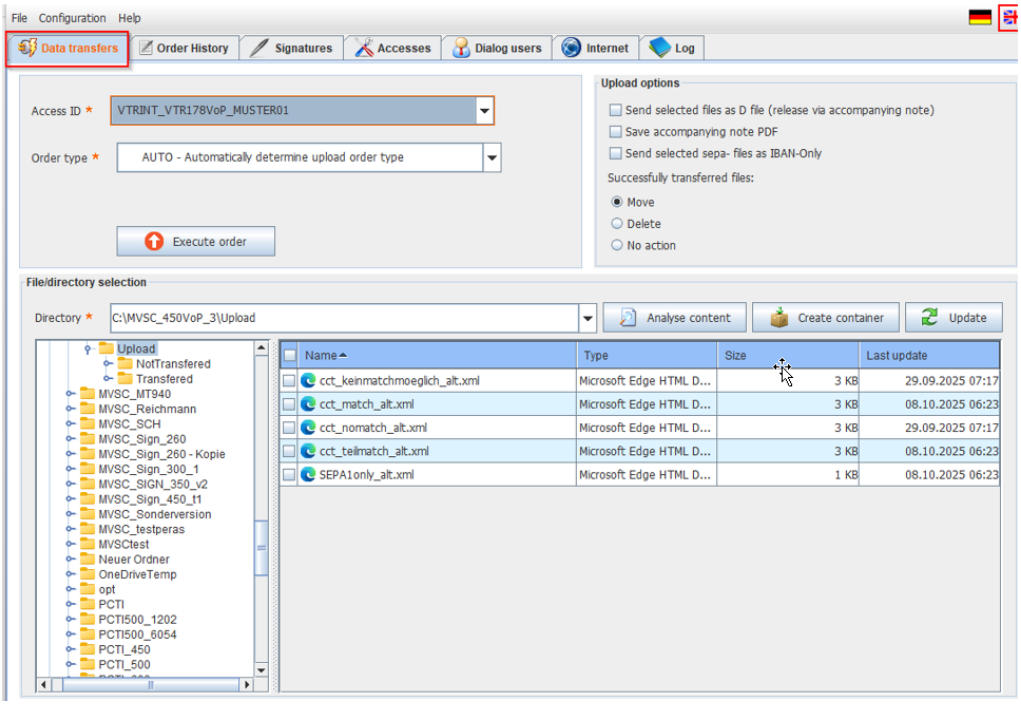


Figure 5.21. VoP: Data transfer with order type 'AUTO'

The user selects the files to be transferred. Then the user clicks on the ‘Execute order’ button.

The order types are then automatically assigned and the files are transferred.

The result is shown in the following illustration:

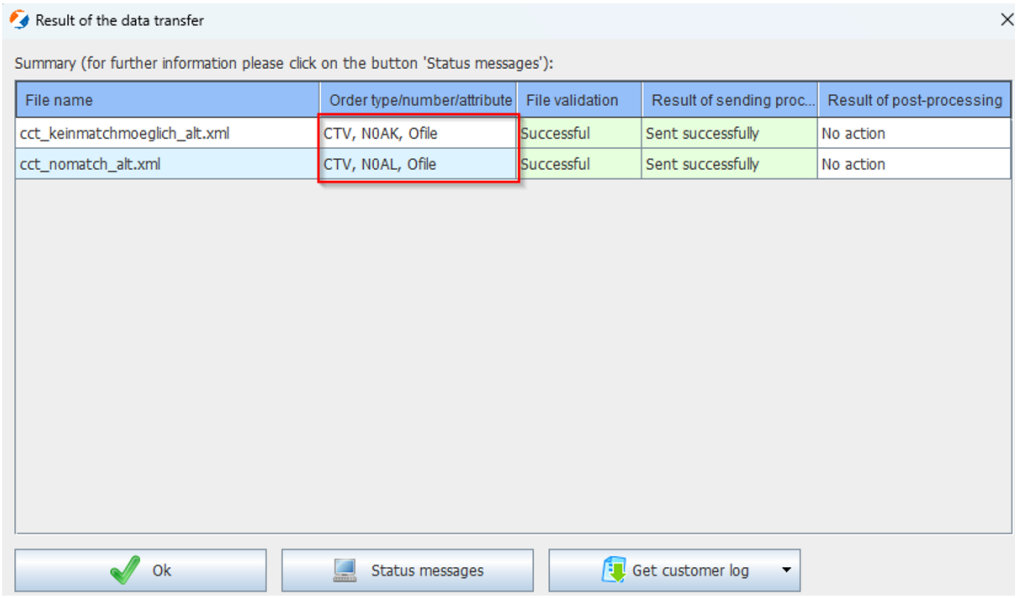


Figure 5.22. VoP: Result of the data transfer with order type 'AUTO'

In this case, only opt-in order types are used automatically.

Data transfer without 'VoP' (opt-Out-request type, i.e. without recipient verification) using the order type 'AUTO':

Precondition: The access includes **de**activating the ‘Enable recipient checking (VoP)’ checkbox in the default settings.

The order type ‘AUTO’ is selected for data transfer.

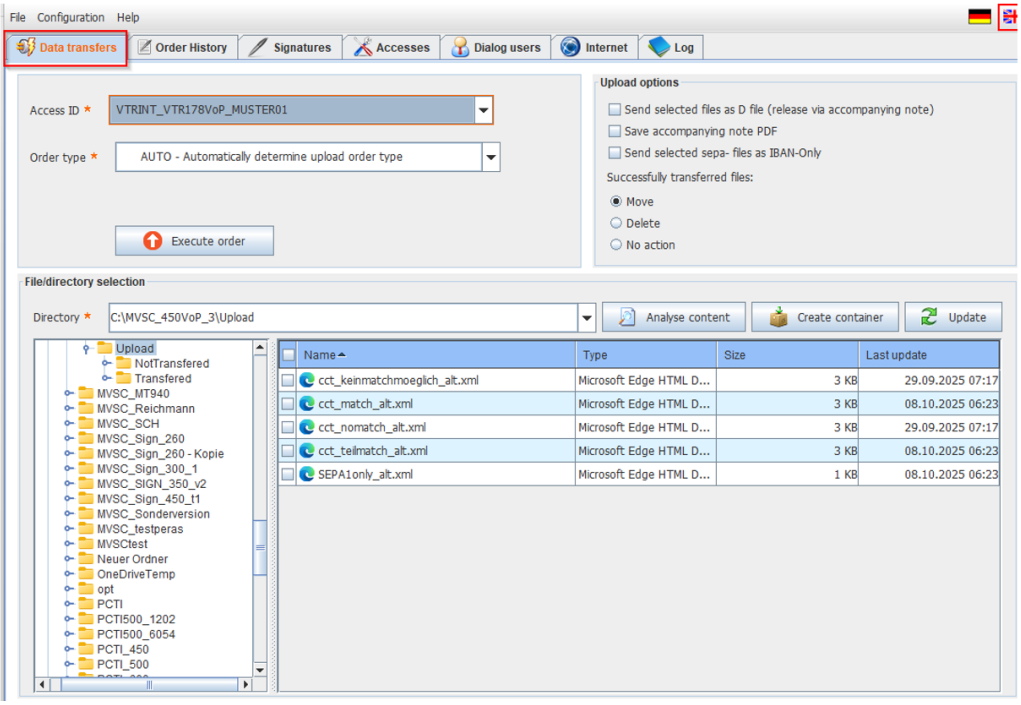


Figure 5.23. VoP: Data transfer with order type 'AUTO'

The user selects the files to be transferred and then uses the button "Execute order".

The order types are automatically assigned and the files are transferred.

The result is shown in the following illustration as an example:

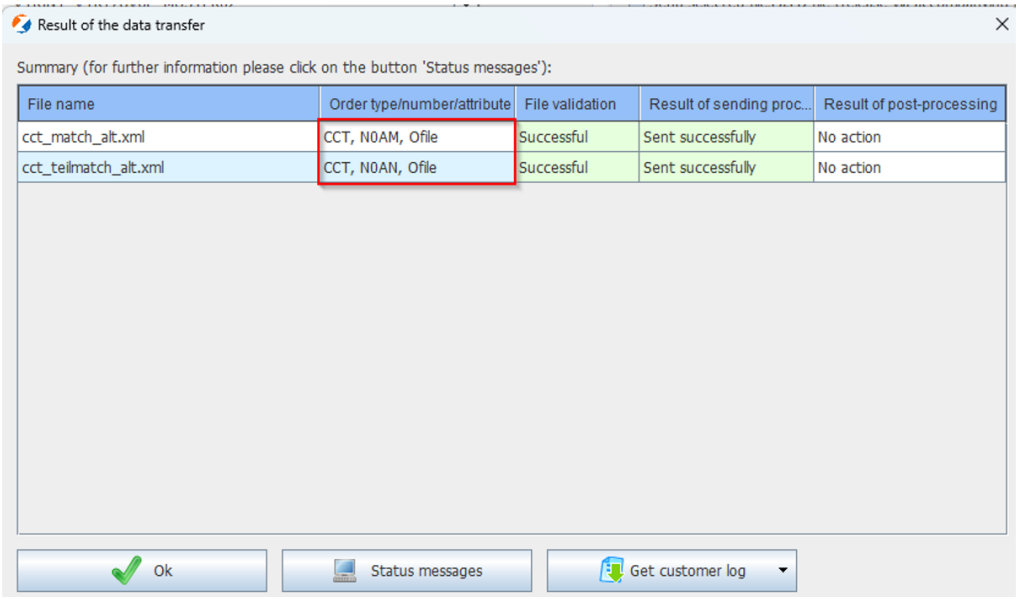


Figure 5.24. VoP: Result of data transfer with order type 'AUTO'

In this case, only opt-out order types are used automatically.

Data transfer with all other order types expect the order type 'AUTO':

A specific order type can also be selected from the order type list. This will then be used regardless of whether it is an opt-in or opt-out order type.

This is illustrated in the following figure:

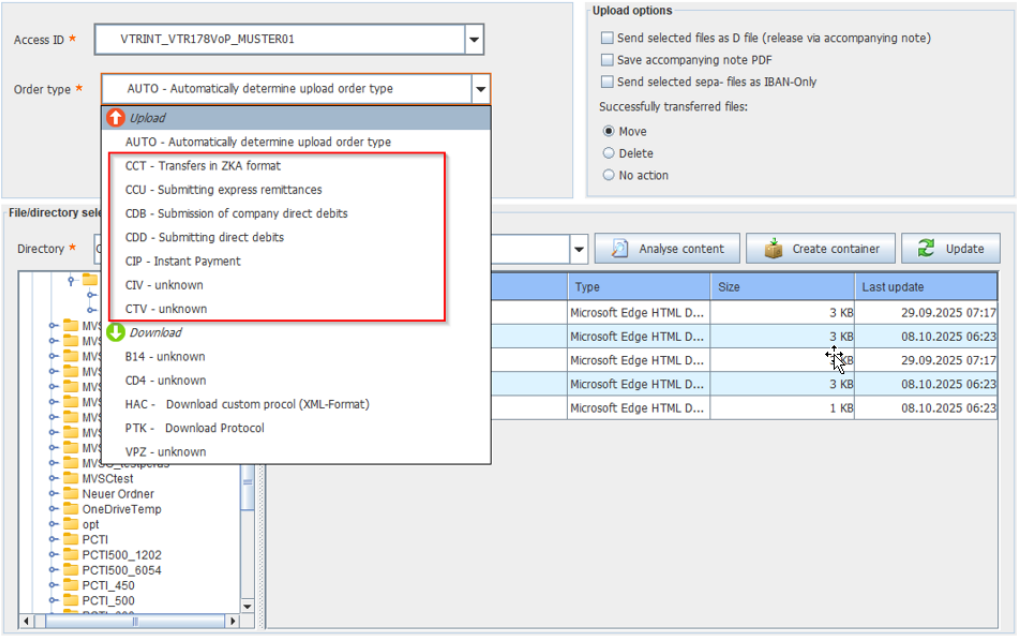


Figure 5.25. VoP: Data transfer with order type other than 'AUTO'

The following table shows the respective opt-out and opt-in order types:

Opt-out order types (i.e. without recipient verification)	Opt-in order types (i.e. with recipient verification)
CCT	CTV
XCT	XTV
CIP	CIV
CCU	-

Opt-out order types (i.e. without recipient verification)	Opt-in order types (i.e. with recipient verification)
XCU	-
CCC	-
XCC	-
XCI	-
CCS	VCS
CIS	VIS
CCX	VCX
CIX	VIX

Information on data transfer with or without recipient verification in batch mode (automatic mode)

Here, too, check that the two checkboxes described in the section above entitled "Default settings":

- Enable recipient checking (VoP) -> relevant for order type 'AUTO'

and

- Validation error output for files with only one job per opt-out order type -> Validation error output for files with only one request per opt-out request type (i.e. without recipient verification)

Data transfer with VoP (opt-in order types) using the order type 'AUTO'

No adjustments need to be made for batch data transfer in the batch job, provided that the order type 'AUTO' is used.

The default settings for the individual entries specify whether the transfer should be carried out using opt-in order types or opt-out order types. These entries are taken into account.

Call with order type 'AUTO':

```
java -jar "MVSC.jar" <$MEINE_ZUGANGSID> AUTO <$MEIN_UPLOADVERZEICHNIS>
```

Data transfer with all other order types except the order type 'AUTO'

However, if qualified opt-out order types are used in the batch job call, you can replace them with opt-in order types if desired.

Example:

If, for example, the previous use of the opt-out order type 'CCT' is replaced by the use of the opt-in order type 'CTV', the call will look as follows:

Previous call with the opt-out order type 'CCT':

```
java -jar "MVSC.jar" <$MY_ACCESS-ID> CCT <$MY_UPLOAD_DIRECTORY>
```

New call with the opt-in order type 'CTV':

```
java -jar "MVSC.jar" <$MY_ACCESS-ID> CTV <$My_UPLOAD_DIRECTORY>
```

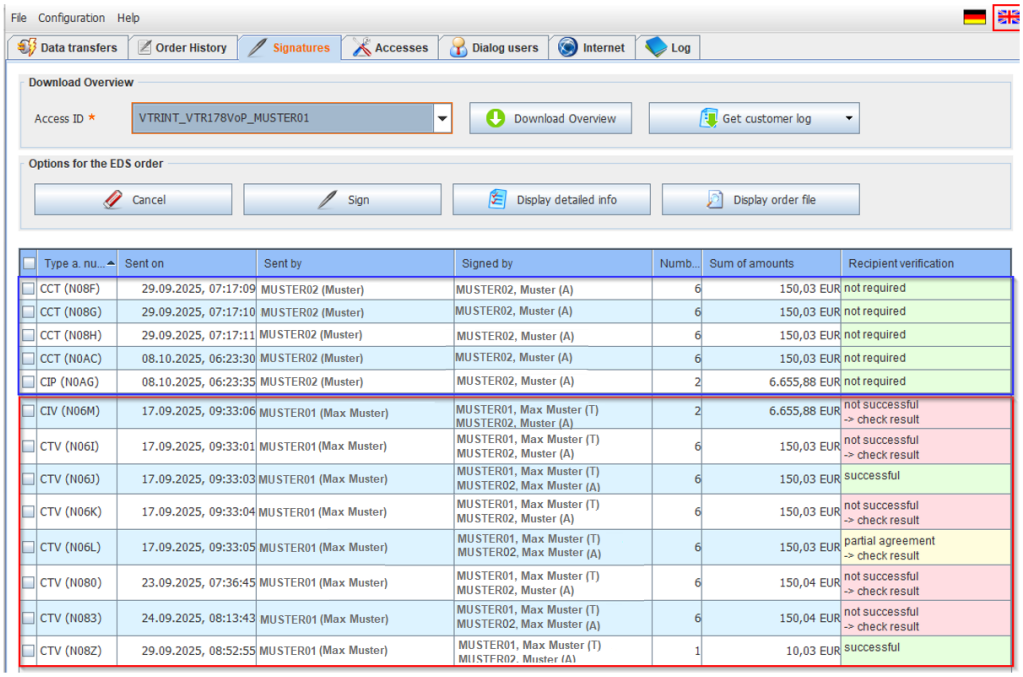
Further processing of orders in the EDS (Electronic Distributed Signature)

Please note the following important information:

Caution
All payment transaction files that are transferred with recipient verification (i.e. with an opt-in order type such as 'CTV') are only stored in the EDS with a **transport signature**. These still need to be signed!

All files to be signed are listed in the download overview.

This download overview is illustrated in the following figure:



Type a. nu...	Sent on	Sent by	Signed by	Numb...	Sum of amounts	Recipient verification
CCT (N08F)	29.09.2025, 07:17:09	MUSTER02 (Muster)	MUSTER02, Muster (A)	6	150,03 EUR	not required
CCT (N08G)	29.09.2025, 07:17:10	MUSTER02 (Muster)	MUSTER02, Muster (A)	6	150,03 EUR	not required
CCT (N08H)	29.09.2025, 07:17:11	MUSTER02 (Muster)	MUSTER02, Muster (A)	6	150,03 EUR	not required
CCT (N0AC)	08.10.2025, 06:23:30	MUSTER02 (Muster)	MUSTER02, Muster (A)	6	150,03 EUR	not required
CIP (N0AG)	08.10.2025, 06:23:35	MUSTER02 (Muster)	MUSTER02, Muster (A)	2	6.655,88 EUR	not required
CIV (N06M)	17.09.2025, 09:33:06	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Muster (A)	2	6.655,88 EUR	not successful -> check result
CTV (N06I)	17.09.2025, 09:33:01	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Muster (A)	6	150,03 EUR	not successful -> check result
CTV (N06J)	17.09.2025, 09:33:03	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Max Muster (A)	6	150,03 EUR	successful
CTV (N06K)	17.09.2025, 09:33:04	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Muster (A)	6	150,03 EUR	not successful -> check result
CTV (N06L)	17.09.2025, 09:33:05	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Max Muster (A)	6	150,03 EUR	partial agreement -> check result
CTV (N080)	23.09.2025, 07:36:45	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Muster (A)	6	150,04 EUR	not successful -> check result
CTV (N083)	24.09.2025, 08:13:43	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Max Muster (A)	6	150,04 EUR	not successful -> check result
CTV (N08Z)	29.09.2025, 08:52:55	MUSTER01 (Max Muster)	MUSTER01, Max Muster (T) MUSTER02, Muster (A)	1	10,03 EUR	successful

Figure 5.26. VoP: Download Overview

In the table illustration, you can see the opt-out order types ‘CCT’ and ‘CIP’ in the first five table rows in the upper section, which are framed in blue. No recipient verifications are carried out for these. Therefore, the right-hand column with the heading ‘Recipient verification’ only states ‘not required’.

No verification by the user is required here.

In the lower section of the last eight table rows, which are framed in red, you can see the opt-in order types. For these, the status of the recipient verification is entered in the right-hand column with the heading ‘Recipient verification’.

There are three possible statuses for this:

1. successfull (with a green background)
2. partial agreement (with a yellow background)
3. not successfull (with a red background)

Detailed information on the individual files can be accessed by double-clicking on the respective row in the table.

For more information on signatures, see the chapter "Data transfer via the GUI", subchapter "Electronic distributed signature".

Retrieve status report and view status file

If you would like more detailed information on recipient verification (‘VoP’) for individual files, you can retrieve and view the status report on recipient verification.

Download status report

To retrieve the status report, select the download order type ‘VPZ’ under the ‘Data transfers’ tab and click on the ‘Execute order’ button.

This is shown in the following illustration:

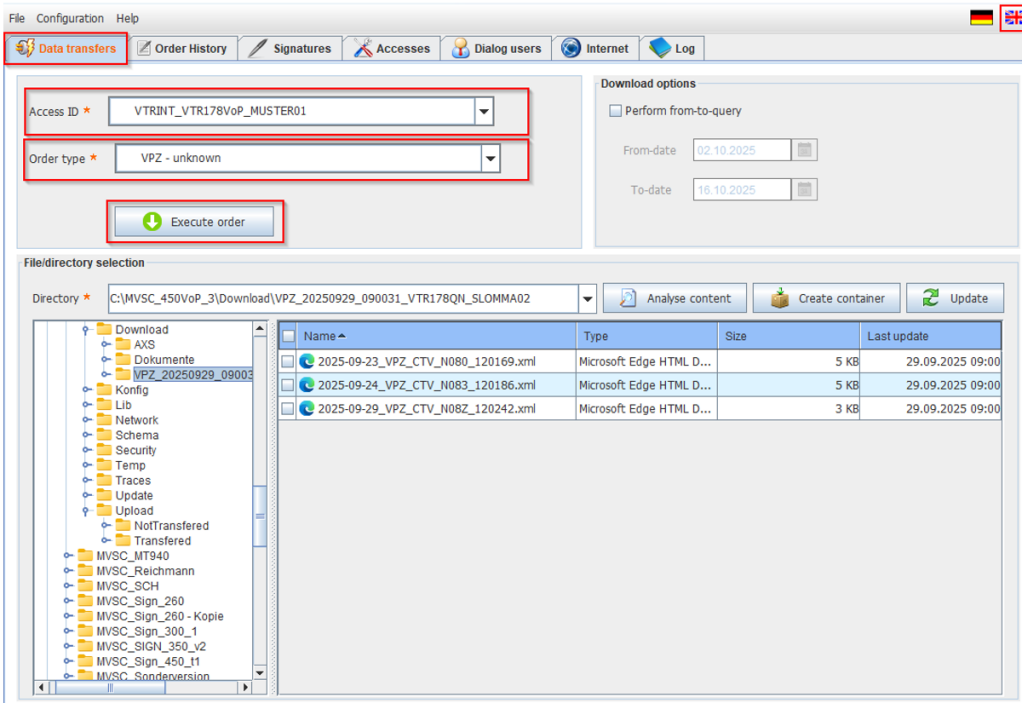


Figure 5.27. VoP: Retrieve the status report, Step 1

In the next step, confirm that you want to unzip the data received in ZIP format by selecting ‘Yes’, as shown in the following illustration.

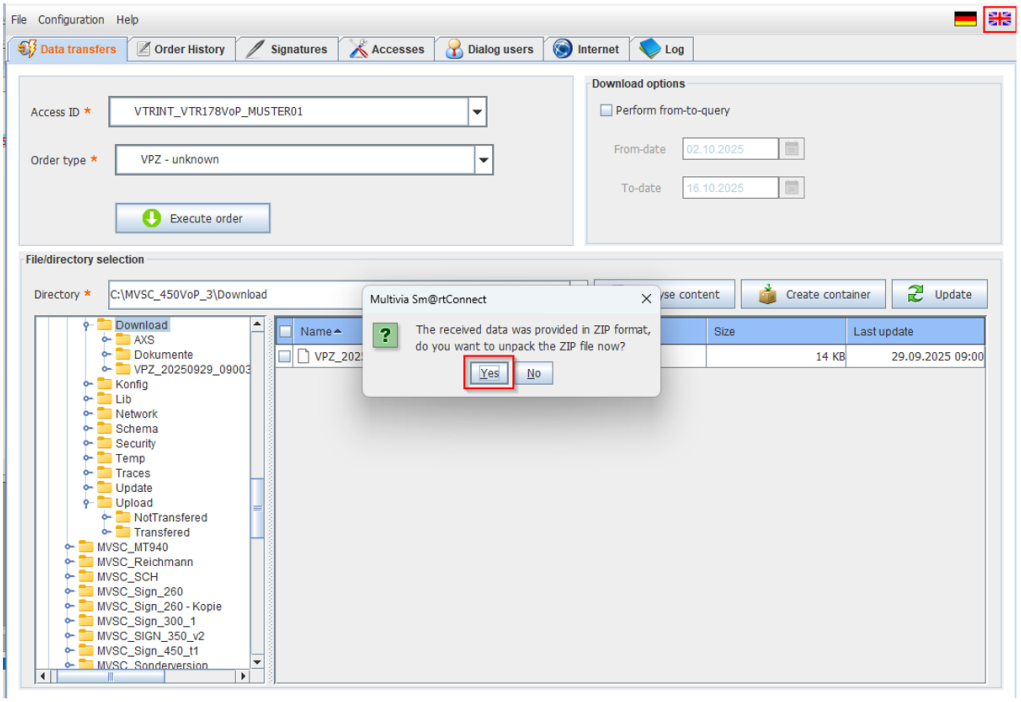


Figure 5.28. VoP: Retrieve the status report, Step 2

The status logs are then listed in a list.

The following figure shows an example of such a list of status logs:

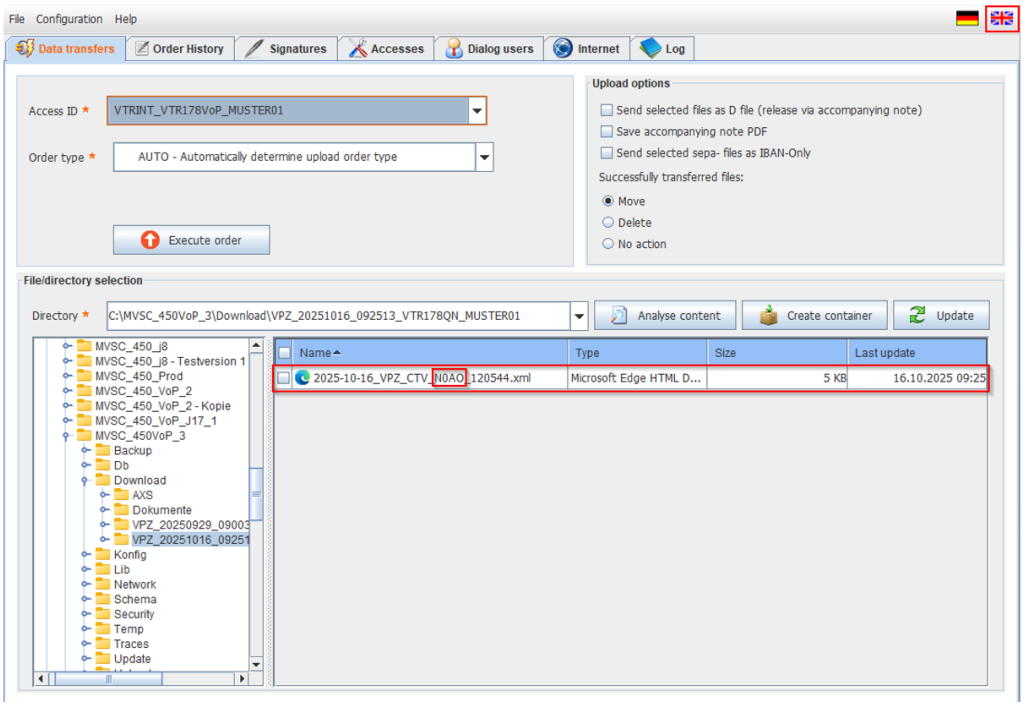


Figure 5.29. VoP: Status logs

Atruvia files can be identified by their order number. For **Atruvia** orders, you will find the order number in the ‘Name’ column in the name after the order type. In the image above, the order number is marked in red.

Example: The name of the job file is ‘2025-10-16_VPZ_CTV_**N0AO**_120544.xml’. The order number is ‘N0AO’.

Display of the VoP status file

The recipient file name verification protocol can now be displayed and saved as a PDF document or be printet.

The following figure shows an example of a status file:

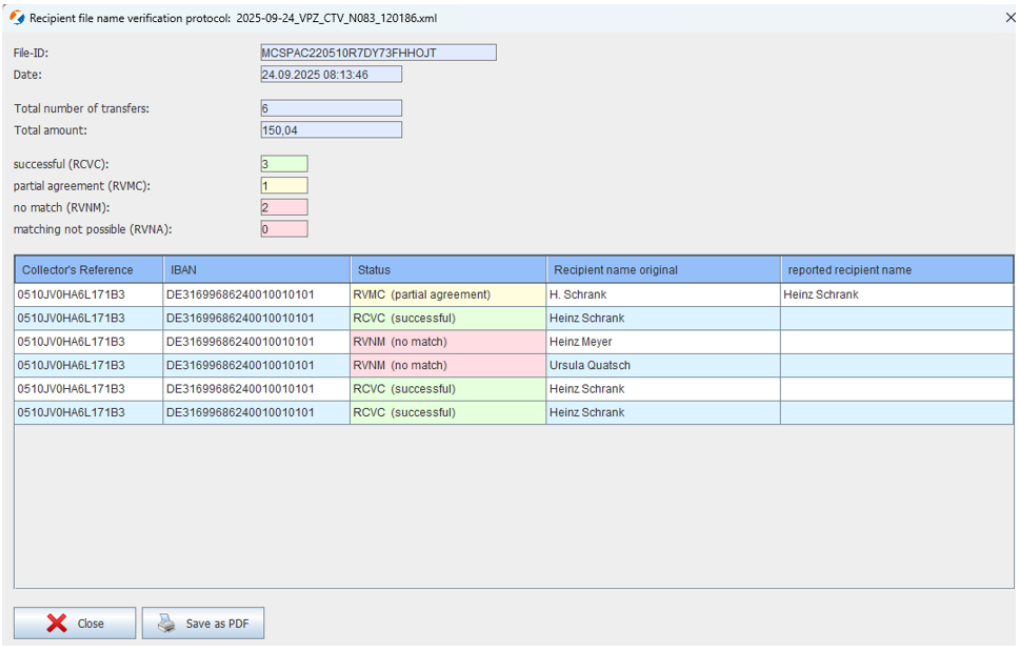


Figure 5.30. VoP: Recipient file name verification protocol

The following figure shows the same status file after sorting by the table column ‘Status’ marked in red:

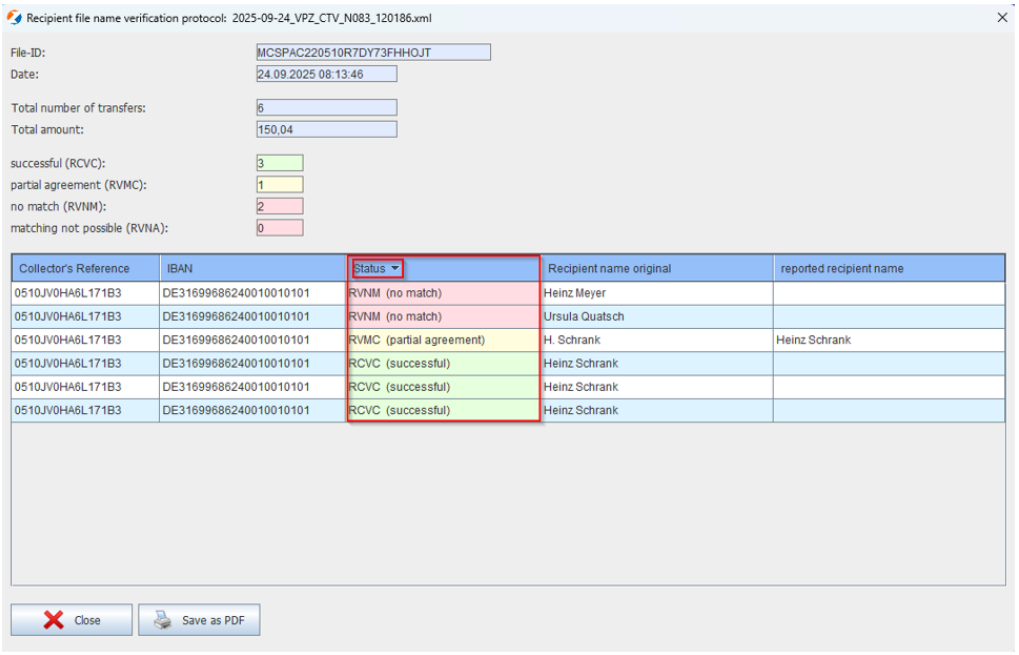


Figure 5.31. VoP: Recipient file name verification protocol sorted by status

Chapter 6. Annex

6.1. File filter

Purpose

File filters play an important role particularly in console mode. For upload orders, they are necessary to filter for the file types that shall be transferred with the configured order type. To this end, the 3-5-digit file extensions that are to be transferred with the respective order type are stored.

Standard settings

Initially, the preset value is the 3-digit identifier of respective order type (e.g. "AZV").

Menu item

The file filter editor can be found via the tab "Accesses". First click on the button "Default settings" and then on the button "File filter for order type".

Validity

The configuration of the file filter can be individually configured for each upload order type per access ID.

Examples

The following table lists some order types with their typical file extensions:

Order type	Typical file extensions
AZV (cross-border payments)	DTAZV
CCT	SEPA XML

6.2. Return values in console mode

General information

If MVSC is used by means of a batch file, the calling application requires information on the result of the transfer process. MVSC returns this information as a numeric value that represents a specific message. The following describes the different return values and their meaning.

Return values and their meaning

If the data transfer was successful, you receive the return value "1". This return value means that the transfer to the bank server was successful for all files found.

If you receive the return value "0", it means that at least one file could not be transferred successfully. In most cases, this is due to formatting errors in the affected file.

Return values lower than zero, on the other hand, indicate an error in the configuration.

The following table lists the possible return values:

Return value	Meaning
25	This return value can occur upon start of the application in console mode. It means that an update is available for MVSC. The update can only be performed in desktop mode.
1	All files found have been successfully transferred to the bank server. Subsequently, the customer protocol (PTK/HAC) has also been successfully downloaded.
0	The data transfer was only partially successful. At least one of the files found was transferred successfully, while other file were not.

Return value	Meaning
	In most cases, formatting errors in the order files are the reason for this return value.
-1	The EBICS access data is incomplete or erroneous.
-2	The Internet access data is erroneous.
-3	The entered order type is not supported or is not activated for this access ID.
-4	No files matching the file filter were found in the specified directory.
-5	At least one of the file found is already being transferred by another dialog user (network installation).
-6	Invalid call parameters were entered.
-7	An access ID that uses a smartcard as signature medium was entered. In console mode, only security files are supported.
-8	A configured/entered directory path is either invalid or you have no write or read authorisation for this directory path.
-9	The application is already running. The data transfer is prevented so that no files are transferred more than once.
-99	You are using a test version that only supports one specific bank sever but not the entered bank server.
-999	Your application version has expired and can no longer be used.

EBICS error messages

All numerical values higher than 1 indicate an issue that was reported back by the EBICS server. These numbers mostly consist of 5 digits and indicate errors that occurred during the EBICS transfer. If such an error occurs, the entire transfer is aborted as the same error can be expected during the transfer of the next file.

The following table lists some of these EBICS return values and their meaning:

Return value/EBICS CODE	Column title
90005/ EBICS_NO_DOWN-LOAD_DATA_AVAILABLE	No data is available for download at the EBICS bank server for this order type. This is not an error as there are several reasons for which no data may be available. For example, it might be the case if the data has already been downloaded.
91002/ EBICS_INVAL-ID_USER_OR_USER_STATE	The sending user is unknown to the bank server or has not been activated in the system yet.
61001/ EBICS_AUTHENTICA-TION_FAILED	There can be different reasons for this error message: • The sending user is not authorised to execute this order type. • The dialog user has not synchronised his private keys (INI) or the public bank keys (HPB) at the bank server yet. • The systems work with different certificates.
91115/ EBICS_OR-DERID_ALREADY_EXISTS	Each order is transferred with its own order number (order ID). It must not be repeated within a specific time period. If this error occurs, two orders with the same order number were sent within a few days. You can fix this issue by adjusting the file "Number.num" in the "config" directory below the installation directory. Open the file "Number.num" and increase the value of the element "BPZ" by at least 2. Please note that the entered value may not exceed the number 46655.

Return value/EBICS CODE	Column title
	Note As of the EBICS version 2.5 this error can no longer occur as the order numbers are allocated centrally at the EBICS bank server.
90003/ EBICS_AUTHORISATION_ORDER_TYPE_FAILED	The submitting user is not authorised for the selected order type. The authorisation for the order type may have been revoked at the EBICS bank server. For the user to receive the order type authorisation, the operator of the EBICS bank server must activate it for that user. For the changes made at the EBICS bank server to become active in MVSC, too, you must click on the button "Download order types" in the tab "Accesses".

6.3. Order types

Order types in EBICS

A distinction is made between upload orders (transfer to the EBICS server) and download orders (download from the EBICS server).

Each order type has a 3-digit alphanumeric ID which is used to unambiguously identify them at the EBICS bank server system. It is also stored at the EBICS bank server, which order format shall be transferred with the respective order types and which business processing shall be performed for the data.

Standard order types

In the EBICS standard, various order types have already been preconfigured for the common order formats.

The following table shows examples for original EBICS order types:

Order type	Transfer direction	Order description	Order format/processing
AZV	Upload	Send AZV in disk format (cross-border payments)	DTAZV
CCT	Upload	Send Credit Transfer Initiation (SEPA credit transfer)	pain.001
CDD	Upload	Send Direct Debit Initiation (SEPA direct debit core)	pain.008
STA	Download	Download SWIFT daily statements (account transactions)	MT940
VMK	Download	Download short-term interim transactions	MT942
C52	Download	Download Bank to Customer Account Report (interim transactions)	camt.052
C53	Download	Download Bank To Customer Statement Report (account transactions)	camt.053

There are more order types predefined by the EBICS standard. They can be found under "http://www.ebics-zka.de/".

6.4. Logging

6.4.1. Dialog user log

Actions of the dialog user

All actions performed by a dialog user in the GUI are saved to a dialog user log. In order for the actions to be traceable chronologically, a new dialog user log is created for each day. The dialog user log does not contain any technical information, but mostly the messages displayed by the application during its use.

Menu item

The dialog user log can be viewed via the tab "Log".

The following figure shows the tab "Log":

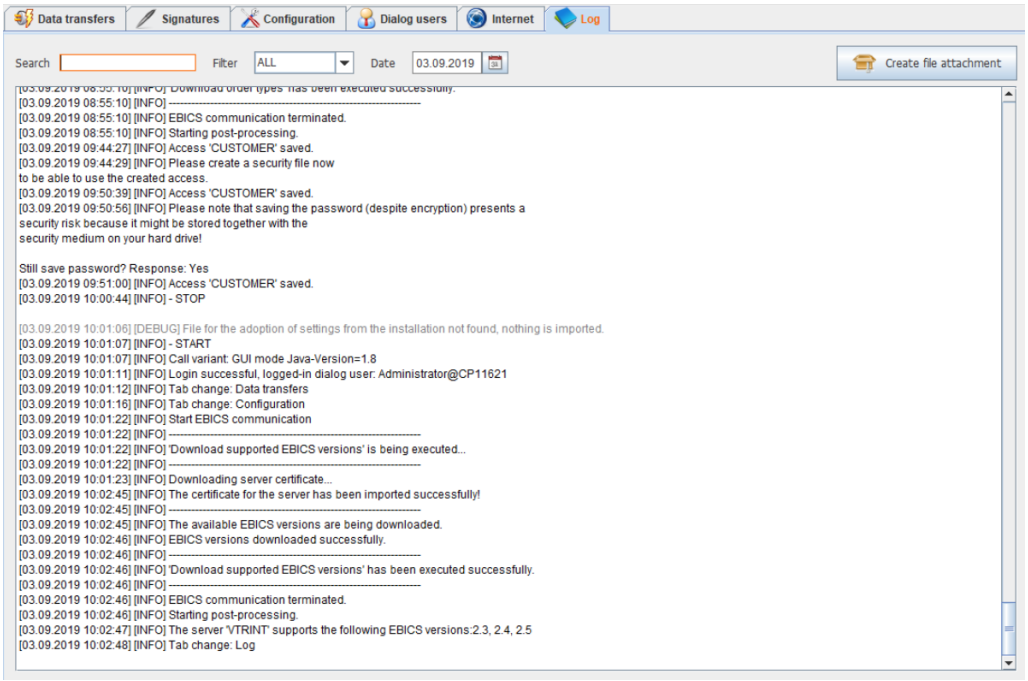


Figure 6.1. Log

Filtering the view

When the tab "Log" is displayed, it lists the latest log entries first. By means of the calendar selection, you can select the day for which the log shall be displayed. The displayed log file can be filtered by certain message types via the selection list "Filter". This way, you can quickly check whether, for example, errors have occurred on a specific day (filter "WARNING/ ERROR").

Additionally, the displayed log can be searched for any words.

Creating a file attachment

Should there be issues with MVSC, the written log files can be exported as a ZIP-compressed file archive via the button "Create file attachment". This archive file can then be sent to the support of Atruvia.

6.4.2. Technical logging

Log level

If persistent issues occur during data transfers that cannot be fixed even after several solution attempts, you can increase the amount of logged data. While the connection is being established, detailed information is recorded which may reveal more about the occurring error.

You can find this setting in the "Help" menu, sub-item "Logging". There are four possible log levels.

These are shown in the figure below:

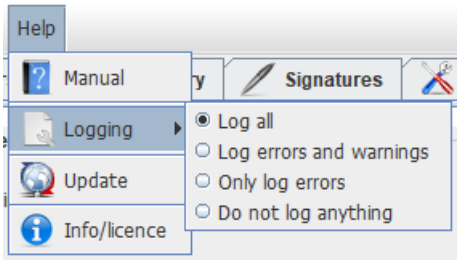


Figure 6.2. Logging

Log files

The log files can be found in the subdirectory "Traces". The files containing the technical logs start with "Trace". The information they contain relates only to the technical process of EBICS data transfers.

6.5. Help

Manual

Under the menu item "Help", "Manual" you can download this manual as a PDF document while using MVSC.