Governikus KG

EGVP

Import interface for 3rd party software

EGVP, Release 3.0.1.0 | Last update January 5th 2017

Table of contents

Legal information and further notes

Although this product documentation was written to the best of our knowledge and with reasonable care, errors and inaccuracies cannot be totally excluded. Legal or other liability for inaccurate information and their consequences cannot be assumed. The information given in this product documentation reflects the current state of development and can be changed without further notice. Future editions can contain additional information. Technical und typographic errors are corrected in subsequent editions.

1 About this manual

This document describes structure and mechanisms of the 3rd party interface of EGVP. The interface offers the dynamic delivery of messages by the specific application that creates messages for the electronic post box. These messages consist of any combinations of message text, cover letter, documents, structured user data for continuing processing or binary content, e.g. pictures and metadata for the respective message.

Reading suggestion

This document addresses members of institutions who want to use EGVP in combination with 3rd party software as well as providers of 3rd party software who want to send or receive messages via EGVP.

- EGVP users find a brief overview about the principles of data transfer between EGVP and other software in chapter "Transferring data to ".
- Providers of third-party software get explanations about how to realize the data transport from and to EGVP and the use of the generic data interface in this specification. There are two possibilities for the exchange of data:
    1. Exchange using the file system
    2. Dynamic transfer of data

Refer to chapters "Interface specification" and "Configuration file govello.txt" for further information.

Additionally, the appendices provide more details about directory and file structures and possible error messages.

Basic comprehension of the technologies below is recommended:

- The standard OSCI transport protocol version 1.2 (refer to www.osci.de)
- OSCI Client Enabler (refer to www.governikus.com)
- EGVP (see www.egvp.de)

More information about EGVP can be found at www.governikus.com and at www.egvp.de.

2 Transferring data to EGVP

The data interface for 3rd party software offers transferring electronic data with all respective parts using a defined structure. EGVP converts the data into an OSCI message and displays this message in the outbox of the local post box.

The messages content data must be presented in a defined structure by the 3rd party software. EGVP imports this data with the help of an integrated component, converting all content data and attachments into the EGVP message format.

After import and conversion of the data, the messages are shown in the EGVP application window (outbox). Now, standard functions like signing and sending can be performed. The interface is identical for all clients.

There are two possibilities for the exchange of data:

1. Exchange using the file system
2. Dynamic transfer of data

Manufacturer's note:

Provided the interface was set up correctly, it will operate continually in the expected manner.

Using the "dynamic" interface the base conditions for every single message can change with the data transfer. When applying a web server there are the following possible restrictions:

- The transfer duration depends on the messages size.
- The speed of the internet connection is as relevant as the web server's performance.
- The use of proxy or firewall can complicate delivery.
- In case of transfer problems the data reload laying on the web server can not be granted.

The "dynamic transfer" interface is suitable for occasional use. For a frequently use of EGVP with many message imports the usage of the "file system" interface is more convenient, since the application client can only be started by invoking the JSP link. Therefore, it is recommended to investigate the exact scenario and use cases before deciding which method to use. When using the dynamic interface, the previously described restrictions (configuration of web server, proxy, and firewall) must be respected in any case.

Note: An XJustiz message has to be attached using the identifier "xjustiz_nachricht.xml". This is the only way to make the message conform to the OSCI transport profile for XJustiz.

2.1 Exchange of data via file system

Functional overview

The data exchange between the respective 3rd party software and EGVP is performed via file system. Both, the 3rd party software as well as EGVP have to define the interface folder once. EGVP users can define this folder for every single post box, needing an interface.

The 3rd party software has to deliver the content data in a defined structure according to the defined specification. A particular EGVP component converts content data and attachments into a EGVP message and provides it in the EGVP post boxes outbox. The user can now finish the message with the EGVP functions (create signatures, encrypt, send).

The interface for 3rd party software is identical for all roles, clients/citizens and public administration/providers.

Limitation for the delivery of data using the transfer folder:

- The exchange of data and the conversion into an OSCI message is not equal to the creation of EGVP messages with the help of the EGVP message dialog.
- The import of 3rd party data is not equal to the import of an OSCI content container.
- The import of 3rd party data will not perform a validation of the content data delivered by the respective 3rd party software. The 3rd party software must make sure that the content data and attachments are correct.
- It is not possible to change the content data of imported messages situated in the EGVP outbox.

Every user, who wants to use the transfer of data via transfer folder, has to establish a post box and to register the post box at the directory service (registration server). Eventually the post box has to be unlocked at the directory service before the user is able to address other recipients. The transfer folder for the delivered content data and attachments has to be defined once in the post box. This directory can be fed by several 3rd party applications and the post box converts all the delivered data into EGVP messages.

Every EGVP post box can only import the data of the one selected transfer folder. The transfer folder has to be chosen via the EGVP user interface before the first use of the interface. It is also possible to set up the data import interval in the respective dialog (at least five minutes).

> **Note**: The German signature law demands that the signer can see the complete and unaltered data before signing.
>
> To offer this possibility, EGVP imports 3rd party messages including content data and puts these messages into the outbox. Here, the user can check and sign the data before sending.

When the user defines the exchange folder, three subfolders will be created automatically:

1. Folder "`Messages`": The provider of your custom software has to place his data in this directory using separate subfolders for every message to import.
2. Folder "`DefectiveMessages`": Messages that cannot be imported because of failures within the data will be transferred from the "Messages" directory into this one. The file log.txt that is as well contained in this folder will contain the error message and an explanation, why the note could not be imported.
3. Folder "`SentMessages`": This folder contains the acknowledgements of all sent messages. Thus the 3rd party software can ascertain whether messages created by the respective software were sent.

2.1.1 Multi-client capability and transfer interface

EGVP offers multi-client capability in a way that the message dialog checks attachments in size and number before creating a message that is ready to send. The 3rd party data interface supports the needed multi-client capability when creating the message and returns error information in case of exceeded file sizes.

2.2 Dynamic delivery of data

In opposition to the exchange via file system, the dynamic delivery of 3rd party data supplies the respective data directly to the EGVP client.

2.2.1 Functional overview

Regardless of the kind of interface, the 3rd party software must deliver the electronic data with all essential information and properties in a defined structure. EGVP creates an OSCI message and stores the message in the outbox.

The dynamic interface differs from the file system interface in a way, that any fix import folder has to be configured. The exchange takes place in a dynamic form instead.

With the help of a configuration file the JNLP link will be created dynamically by the web server. The configuration file contains all necessary information for the data transfer including where to find content data and attachments. EGVP uses this information to load the data and create an OSCI message.

After successful import and conversion the message is displayed within the application window of the electronic post box, that is, the outbox. The message can be handled with the standard client functions, like sign and send. The interface is available for every role, and it is identical for all clients.

Limitation for the dynamic delivery of data:

- The dynamic interface is not identical to the OSCI message import of messages created by EGVP.
- The dynamic interface cannot be equalised with the import of an OSCI content container.
- The dynamic interface will not perform validity checks of content data, delivered by the 3rd party software. The correctness of content data and attachments must be assured by the 3rd party software.
- It is not possible to edit imported messages within the outbox. The message will be displayed in the usual way with the respective tabs for details.
- It is possible to import several messages at the same time.
- There is no direct communication between EGVP and the involved 3rd party software.
- The 3rd party software has to provide all necessary data. The web server will generate the JNLP link dynamically to start EGVP. The provided data is handled by the 3rd party software, thus the 3rd party software is responsible for deleting or saving the data after import.
- The 3rd party data/the web server has to initiate the following processes:
    - Provide all necessary data for the download by the 3rd party software,
    - Dynamically generate the JNLP-file by the web server,
    - Launch the EGVP client,
    - Create an OSCI message using the loaded data by EGVP, and
    - Provide the OSCI message within the outbox by EGVP.

Note: The German signature law demands the possibility to see the complete and unaltered message data before signing. Thus EGVP converts the 3rd party data into the outbox, where it can be verified and signed before sending.

Note: In case the transfer of data (configuration file, content data or attachments) from a web server or file system was not successful, e. g. because of a missing connection to the download server, an error message is displayed and the process is aborted.

Possibly the outbox message can be loaded completely with the context menu's function "Re-Import". Incorrect messages are marked red and have a special status symbol in the message area.

3 Interface specification

The data interface for 3rd party software offers the possibility to transfer electronic data with all respective parts using a defined structure. EGVP converts the data into an OSCI message and displays this message in the outbox of the local post box.

The message's content data must be presented in a defined structure by the 3rd party software. EGVP imports this data with the help of an integrated component, converting all content data and attachments into the EGVP message format.

After import and conversion of the data, the messages is shown in the EGVP application window (outbox). Now, standard functions like sign and send can be performed. The interface is identical for all clients.

There are two possibilities for the exchange of data:

1. Exchange using the file system
2. Dynamic transfer of data

Manufacturer's note:

Provided the interface was set up correctly, it will operate continually in the expected manner.

Using the "dynamic" interface the base conditions for every single message can change with the data transfer. When applying a web server there are the following possible restrictions:

- The duration of transfer depends on the messages size.
- The speed of the internet connection is as relevant as the web server performance.
- The use of proxy or firewall can complicate the delivery.
- In case of transfer problems the data reload laying on the web server can not be granted.

The "dynamic transfer" interface is suitable for occasional use. For a frequently use of EGVP with many message imports the usage of the "file system" interface is more convenient. Therefore, it is recommended to investigate exact scenarios and use cases before deciding which method to use. When using the dynamic interface the previously described restrictions (configuration of web server, proxy, and firewall) must be respected in any case. Both kinds of data interface are described below.

3.1 Exchange of data via file system

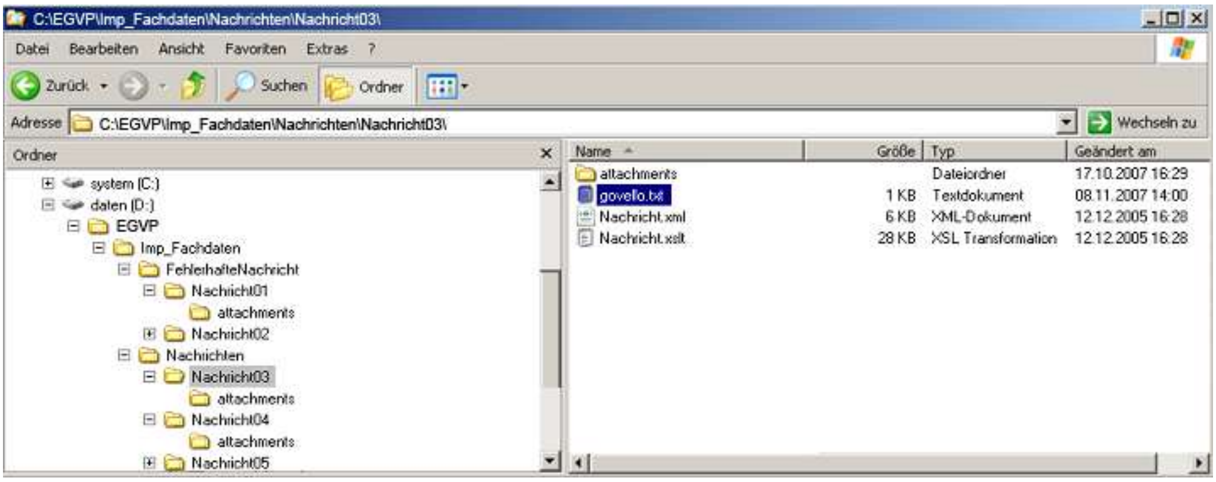
Functional overview

The data transfer between EGVP and 3rd party software can take place via the file system. 3rd party software as well as EGVP has to define an exchange directory once. EGVP users can define this exchange directory for each P.O.box in the user interface of EGVP. Limitations for the data transfer with the exchange directory:

- The import of 3rd party data will not perform a validation of the content data delivered by the respective 3rd party software. The 3rd party software must make sure that the content data and attachments are correct.

3.1.1 Specification and configuration of the exchange directory

When the user defines the exchange folder, three subfolders are created automatically:



C:\EGVP\Imp_Fachdaten\Nachrichten\Nachricht03\

```
⊞ system (C:)
⊟ daten (D:)
  ⊟ EGVP
    ⊟ Imp_Fachdaten
      ⊟ FehlerhafteNachricht
        ⊟ Nachricht01
            attachments
        ⊞ Nachricht02
      ⊟ Nachrichten
        ⊟ Nachricht03
            attachments
        ⊟ Nachricht04
            attachments
        ⊞ Nachricht05
```

Name	Größe	Typ	Geändert am
attachments		Dateiordner	17.10.2007 16:29
govello.txt	1 KB	Textdokument	08.11.2007 14:00
Nachricht.xml	6 KB	XML-Dokument	12.12.2005 16:28
Nachricht.xslt	28 KB	XSL Transformation	12.12.2005 16:28

Figure 1: Exchange directory for business data

Folder "Messages"

The message folder is the main directory where all messages generated by the 3rd party software are saved.

The folder "`Messages`" contains a file "userid.txt" created by EGVP, which includes the clear user's identification number, which is assigned by the registration server.

The file `userid.txt` and the message folder "`Messages`" are created by EGVP, not by the 3rd party software.

For every message in this directory a subfolder (*Nachrichtenordner*) has to be created and labelled with **a unique and distinct** name. The 3rd party software has to place the content data of the respective message in this folder.

If additional documents are assigned to the message, a subfolder "attachments" has to be created in the respective message folder and the documents have to be placed here.

Message folders, which were not moved to the directory "DefectiveMessages" or deleted after successful import, will be locked.

"Locked" in this case means that a file "import.lock" is saved in the respective message folder.

	Note: If EGVP is not able to move or delete a message folder this can be caused by another application accessing files in the folder.

During the next import attempt of a new message EGVP will try to move or delete the message folder.

Folder "DefectiveMessages"

This is the folder where incorrect messages are moved to when an error occurred during the exchange or conversion.

The folder ("`DefectiveMessages`") is created by EGVP, not by the 3rd party software.

Additionally EGVP creates a file log.txt where all imports are logged. It is continued with every failed import.

Folder "SentMessages"

In this directory acknowledgements for sent messages are provided for the 3rd party software. For each sent message created by a 3rd party software a subfolder is created and an acknowledgment in HTML format is saved there.

In case the 3rd party software has defined its own message-ID, it is used as folder name for the special sub folder. If no specific value is defined in the configuration file for the parameter `message_id`, the usual unique message ID is used.

In the following the given structure of the exchange directory is shown. For the hierarchical structure see Figure 1: Exchange directory for business data

Element	Explanation
<exchange directory>	Has to be defined as exchange directory by the EGVP user.
Messages	Is created by EGVP, directory name is not alterable.
<message folder>	Is created by the 3rd party software; name is alterable; unlimited number of directories can be created, names have to be unique and clear.
attachments	Is created by the 3rd party software; directory name is not alterable.
<attachment.xyz>	Is created by the 3rd party software; name is alterable; number and size of attachments depends on the client configuration
govello.txt	Is created by the 3rd party software; name is not alterable.
<inhaltsdaten.xml>	Is created by the 3rd party software; name is alterable.
<inhaltsdaten.xsl>	Is created by the 3rd party software; name is alterable.
<inhaltsdaten.xyz>	Is created by the 3rd party software; name is alterable; unlimited number of documents can be defined as content data.
userid.txt	Is created by EGVP, file name is not alterable.
DefectiveMessages	Is created by EGVP, directory name is not alterable.
log.txt	Is created by EGVP, file name is not alterable.
SentMessages	Is created by the software client; directory name is not alterable.

Element	Explanation
<message folder>	is created by the software client, directory name corresponds to the message ID from the 3rd party software or is the unique message-ID of the message.
acknowledgement.html	Is created by the client, file name is not alterable.

Figure 2: Structure of the directory of the import interface

3.1.2 Content data and attachments

The 3rd party software can determine that the content of a message is visualised in the "message" tab. Therefore an XML-file and an XSL-file must be defined in the `govello.txt`. These are converted to an HTML-page during the import process and are shown in the "message" tab.

All files saved in the respective message directory, except for the configuration file `govello.txt`, are considered content data in terms of OSCI and are imported accordingly. All files in the subfolder `attachments` are handled as attachments in terms of OSCI.

Note for use of special characters in attachments and content data:

The following characters are accepted in the filenames of attachments and content data:

- All characters of the alphabet (upper and lower case)
- All numbers
- Special characters: ! § $ % ( ) = , . - _ + # ' @
- Blanks

3.1.3 Configuration file govello.txt

For every message EGVP needs meta-information, which has to be provided in the file `govello.txt`. The structure is described in the chapter "Configuration file govello.txt". It is mandatory that the file is named "govello.txt".

The configuration file `govello.txt` should be the **last** file to be placed in the specific message directory. This helps avoiding problems during import.

Note: If the automated import of messages is active, problems during the import process can occur, in case messages were not completely transferred to the message directory. If the first import attempt fails EGVP starts a second attempt.

3.1.4 Multi-client capability and transfer interface

The logic of the multi-client capability of the EGVP message window is established in the transfer interface as well. The transmission of a message is limited concerning the number of attachments and the overall size for all roles.

By default, the maximum file size is 30 MB and the maximum number of attachments is 100.

3.2 Dynamic transfer of data

Dynamic transfer of data means that the transmission of the data takes place directly to EGVP instead of using the file system.

3.2.1 Functional overview

Similar to the data transfer via the exchange directory, the 3rd party software has to provide the data with all of its components in a certain structure for EGVP to create an OSCI message for the outbox.

The difference between the dynamic interface and the exchange directory is that no specified import and export directories have to be defined but the transfer takes place dynamically.

With a dynamically generated JNLP and the included link to the configuration file all necessary information for a direct delivery of data is provided.

The data, which should be transferred, has to be kept ready on a web server or the file system. EGVP downloads data via HTTP/HTTPS or from the file system.

All information necessary for the addressing (as well as information for further processing) is transferred in the configuration file govello.txt. EGVP creates an OSCI message after successful transfer. After transfer and conversion the message is shown in the outbox of the P.O.box and is ready for further handling.

Limitation for the dynamic transfer of data:

- The dynamic interface is not identical to the OSCI message import of messages created by EGVP.
- The dynamic interface cannot be equalised with the import of an OSCI content container.
- The dynamic interface will not perform validity checks of content data, delivered by the 3rd party software. The correctness of content data and attachments must be assured by the 3rd party software.
- It is not possible to edit imported messages within the outbox. The message is displayed in the usual way with the respective tabs for the details.
- It is possible to import several messages at the same time.
- There is no direct communication between EGVP and the involved 3rd party software.
- The 3rd party software has to provide all necessary data. The web server will generate the JNLP link dynamically to start EGVP. The provided data is handled by the 3rd party software, thus the 3rd party software is responsible for deleting or saving data after import.
- The 3rd party data/the web server has to initiate the following processes:
  - Provide all necessary data for the download by the 3rd party software,
  - dynamic generation of the JNLP-file by the web server,
  - launching the EGVP client,
  - creating an OSCI message using the loaded data by EGVP, and
  - providing the OSCI message within the outbox by EGVP.
- If multiple users access the same application folder on a network, a multiple start of the application client by invoking a JSP link is not possible.

> **Note**: In case the transmission of a file (configuration file, content data and attachments) from a web server or the file system was not successful (e.g. because of missing connection to the download server) a corresponding message is shown and the process is aborted.
>
> If the transmission was not successful the incomplete message is marked red in the outbox and also has a special symbol. The import can be repeated with the "re-import" function from the context menu.

3.2.2 Detailed specification

General requirements

Users select the data which should be transferred in a message with a browser or a 3rd party software (content data and attachments). The data is stored on a central download server or its file system. This is controlled by the 3rd party software and the 3rd party software has to ensure that the data is available for a later upload. In addition to these data the 3rd party software has to create a configuration file and also store it on a central download server.

The browser/3rd party software automatically starts further processing. The launch is always carried out by a dynamically generated JNLP-file, which is generated by the web portal. Therefore the 3rd party software launches a JSP-file, which is made available by the provider. The JNLP-file contains all necessary information (URL to configuration file with the parameter `config.file`) to begin further processing.

Feedback for 3rd party software

After the import interface is configured the acknowledgement for sent messages is placed in the folder `SentMessages` in the defined exchange directory. With the parameter `message-ID` in the configuration file a directory name for the acknowledgement can be set.

3.2.3 Flowchart

A rough outline of the process is shown in the following figure:

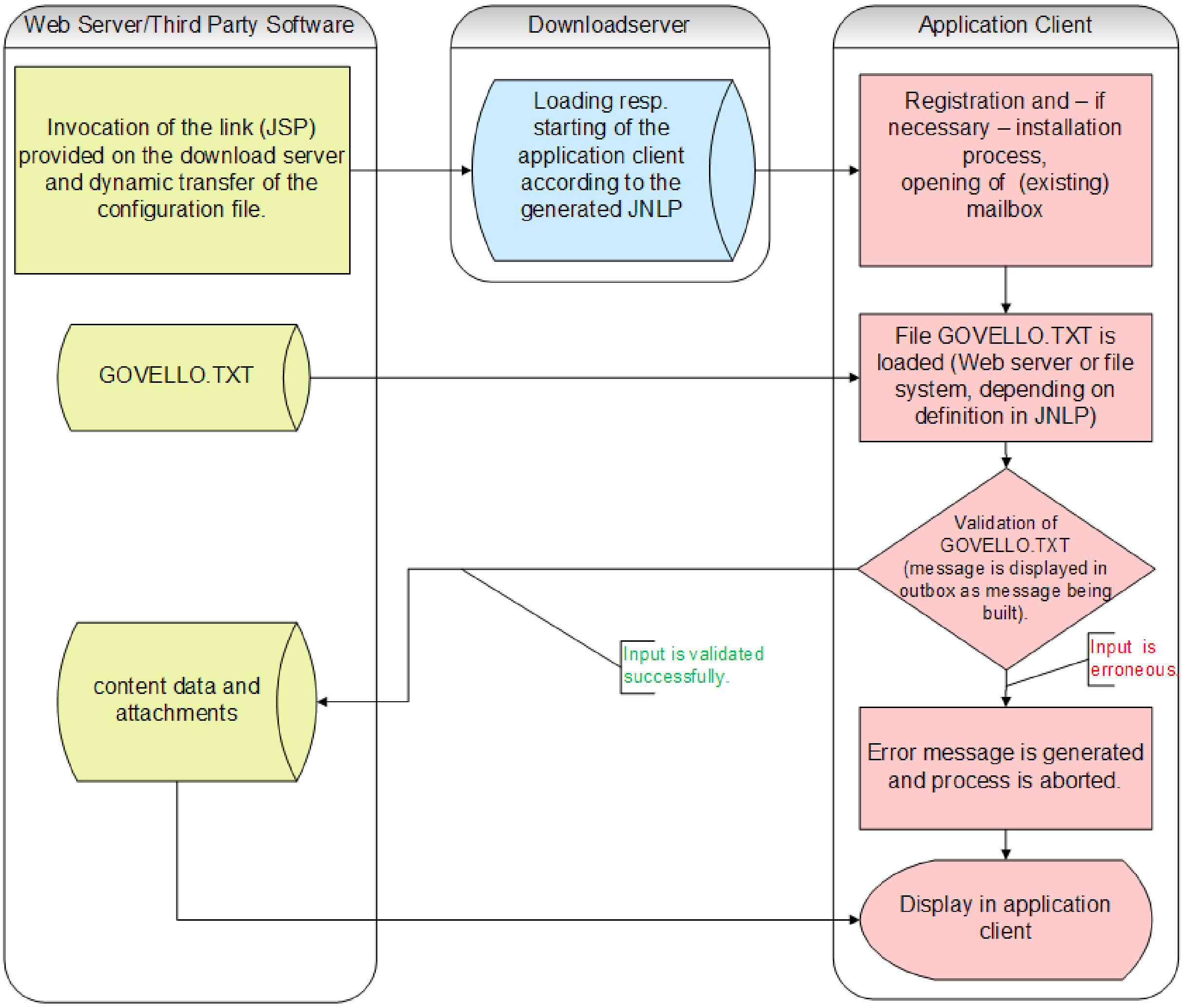

Web Server/Third Party Software
Downloadserver
Application Client
Invocation of the link (JSP) provided on the download server and dynamic transfer of the configuration file.
Loading resp. starting of the application client according to the generated JNLP
Registration and – if necessary – installation process, opening of (existing) mailbox
GOVELLO.TXT
File GOVELLO.TXT is loaded (Web server or file system, depending on definition in JNLP)
Validation of GOVELLO.TXT (message is displayed in outbox as message being built).
Input is validated successfully.
Input is erroneous
content data and attachments
Error message is generated and process is aborted.
Display in application client

Figure 3: Outline and flowchart of the dynamic data transfer interface

The steps during the transfer of data via the dynamic transfer interface are:

1. Starting the application from a download server: Invocation of the link (JSP) on the download server including dynamic transfer of the configuration file. JSP enables the generation of a dynamic JNLP.
    a. D:\message\govello.txt
    b. http://10.211.1.70:8888/egvp2/message/govello.txt

	Note: The JSP depends on the EGVP client, which should be started. Please ask your operator. This JSP-Link has to be extended by the parameters for the configuration file: <http-path>/client.jsp?configfile=<config file>

2. EGVP is started or downloaded if there is no client installed. If a EGVP-client is already running no second client is started.
3. The login process (if necessary, the installation) of EGVP is passed and an existing P.O.box is opened/established.
4. The govello.txt (chapter Configuration file govello.txt)is downloaded from the web server.
5. The govello.txt is validated. The message is shown in the outbox as message built construction.

   a. If any entries are incorrect an error message is generated and the process is terminated. The error message is shown in EGVP.

   b. If the entries addressee-ID, signature level and message type are successfully validated, the processing will continue.

6. The content data and attachments are downloaded consecutively from the web server. Afterwards, the message is in the outbox and can be signed and sent.

3.2.4 Checks before EGVP is started

Before a 3rd party software starts EGVP it should check whether EGVP is installed. If EGVP is already installed it should be checked if the software is running properly and can be addressed directly.

1. Is EGVP installed? To check if EGVP is already installed Java Web Start mechanisms are used. In addition, a 3rd party application can check for an installed post box. Therefore the 3rd party software can check for the existence of EGVP system files. The following files are a sign for the existence of a EGVP installation:

   a. govello20.properties: The file can be found in the user directory of the operating system of each user with an installed P.O.box. The file informs about the location the OSCI-directory is stored at.

   b. OSCI directory including subfolders: Within the OSCI directory there are different subfolders. A subfolder EGVP_xxx with additional subfolders postfach_mailbox_xxx indicates for the 3rd party software that the user has created a P.O.box.

You can find additional information about the file deposition in the chapters "Appendix 3 " and "Appendix 4 Message directory" The firmly-defined file deposition gives information on how directories and data of the P.O.box are organised in the file system.

2. Is EGVP already started? While transferring data EGVP ascertains by a lock file stored in the user folder, if the P.O.box is already opened by a user. Consequently, the 3rd party software is freed from checking this. There is no direct connection between EGVP and the 3rd party software. There are two different scenarios of a running EGVP to be considered.

   a. EGVP is started and a P.O.box is not open

   b. EGVP is started and a P.O.box is open

If no P.O.box is open, the selection dialog for P.O. boxes is displayed.

If EGVP is not running it is started automatically by invoking the JSP link. In case a lock file exists, no additional start of the application client is executed.

The reaction to certain check results is described in detail in "Appendix 2 Scenario".

3.2.5 Content data and attachments

All files defined in the configuration file as `attachments` are handled as attachments in terms of OSCI.

The 3rd party software can determine that the content of a message is visualised in the "message" tab. Therefore an XML-file and a XSL-file must be defined in the `govello.txt`.

These are converted to an HTML-page during the import process and are shown in the "message" tab. The XML-file and XSL-file are content data in terms of OSCI.

Note for use of special characters in attachments and content data:

The following characters are accepted in the file names of attachments and content data:

- All characters of the alphabet (upper and lower case)
- All number
- Special characters! § $ % ( ) = , . - _ + # ' @
- Blanks

4 Configuration file govello.txt

For each message EGVP needs certain meta-information. These are deposited in the `govello.txt`. The build is described in the following. The file name `govello.txt` may vary with the dynamic interface as it is handed over as a parameter.

The file contains parameters, which EGVP needs to handle addressing, visualisation of the message, the signature level and assignment of attachments. All parameters have to be in "Key/Value pairs" and are case sensitive.

Structure of the configuration file

The following listing shows the elements the configuration consists of:

Field name	**Data type/ Length**	**Mandat./ Opt.**	**Value**	**Content/Explanation**
type	String/ 255	mandatory	List see appendix	Defines the type of EGVP message. Possible message types are listed in a firmly defined listing. The listing can be found in the appendix.
id	String/ 255	mandatory		The ID defines the addressee. It can have different values.
			any according to explanation	This ID is the ID of the registration server, which is assigned to every user (GOVELLO ID)
			any according to explanation	This ID is the ID of the registration server respectively of the local address book, which is assigned to a distribution list. (DIST-ID).
			any according to explanation	With the administration console of the registration server for each user a clear number/key can be assigned beside the EGVP user-ID (e.g. for providers the XJustizID). Because addressing within the EGVP system is made by the user-ID it is necessary to link user-ID and user-defined ID. This is made with the parameter "key".
key	String/ 255	mandatory		Defines the value the ID is connected to the user data. There a two possibilities. Upper and lower case has to be kept for the keys!
			USER_ID	Corresponds to the User-ID assigned by the registration server.
			DIST_ID	Corresponds to the distribution list-ID (DIST-ID) which is assigned locally or by the registration server.
			ROLE	Corresponds to the attribute "Externe-ID" of the user and is assigned by the administration tool of the registration server.
signlevel	String/	mandatory		Defines the required signature level. All values are case sensitive!

Field name	Data type/ Length	Mandat./ Opt.	Value	Content/Explanation
	1		o	No signature
			f	Advanced signature
			q	Qualified signature
subject	String/ 255	optional	any text	3rd party software can set the subject, which is shown in the message area.
attachments	String	optional		**Only** to be set for dynamic data transfer: Defines the URL to the storage of attachments. All files defined as `attachments` in the configuration file are handled as attachments in terms of OSCI. Several attachments are separated by a semicolon. In the URL the exact file name has to be specified.
nachricht	String/ 255	optional	file name	If the message should be visualised in the tab "message" a valid XML-file has to be defined here. a) Exchange via file system: The name of the XML-file is changed to nachricht.xml after import and the file is sent under this name as content data in terms of OSCI. Changing the name helps the addressee to differentiate messages. If data is transferred via the exchange directory in the file system the name of the XML-file can be specified here. The XML-file is transferred to the OSCI message as content data and has to be stored with the same name in the message folder. b) Dynamic data transfer: If data is transferred to EGVP dynamically the URL to the storage of the XML-file can be defined here. The XML-file is included in the OSCI message as content data. The file has to be stored with the same name in the download area/the file system.

Field name	Data type/ Length	Mandat./ Opt.	Value	Content/Explanation
message_id	String/ 255	optional	any text	Defines the directory name for confirmation to the 3rd party software. After the message was sent successfully the acknowledgement of receipt is stored in the folder `SentMessages`. The directory name is defined by the message_id. The default is the message-ID of the sent message.
stylesheet	String/ 255	optional	file name	If the message should be visualised in the tab "message" a path to a valid XSLT-file has to be defined here. In conjunction with the XML-file EGVP converts it to an HTML-file. The name of the XSLT-file is changed to nachricht.xsl after import and the file is sent under this name as content data in terms of OSCI. Changing the name helps the addressee to differentiate messages. a) Exchange via file system: If data is transferred to EGVP via the exchange directory the name of the XSLT-file can be defined here. The XSLT-file is included in the OSCI message as content data. The file has to be stored under the same name in the message folder. b) Dynamic data transfer: If the data is transferred dynamically to EGVP the URL to the storage directory of the XSLT-file can be defined here. The XSLT-file is included in the OSCI-message as content data. The file has to be stored under the same name in the message folder.

The value of optional fields that are given in the govello.txt has to exist as defined. E.g. if the XML-file is specified but does not exist, the message is shown as not completely imported.

Note:

Attachments are not referenced in the govello.txt, but they are included in the message container in the folder "attachments".

1. Example (exchange directory)

```
type=Allgemeine Nachricht
id= govello-1207317030812-000000057
key=USER_ID
subject=Information
message=nachricht.xml
stylesheet=nachricht.xsl
siglevel=q
message_id=MeineSoftware_4711_001
```

Figure 4: Example for a configuration file govello.txt for the data transfer via exchange directory

2. Example (dynamic transfer):

```
type=Govello Message
id=B1100
key=ROLE
subject=Anfrage
message=http://xyz/message.xml
stylesheet= http://xyz/message.xsl
signlevel=q
attachments=http://xyz.doc;http://abc.txt;http://efg.pdf;...
```

Figure 5: Example for a configuration file `govello.txt` for dynamics data transfer

Appendix 1 Definition of message types

All message types in the table below can be selected in the message dialog.

Message type	Use for
Allgemeine Nachricht	All messages
Testnachricht	System or functional tests.

Appendix 2 Scenarios

In the following different scenarios, which support the dynamic data transfer, are illustrated. It is depicted what has to happen if the application client:

1. is not installed or
2. is installed but not started or
3. is installed and started.

Finally the data transfer to the P.O.box is illustrated.

Scenario 1 – The application client is not installed on the user's computer

After the download initiated by the 3rd party software is completed by Java Web Start the common installation process will start. The user is asked to

- agree to the installation of the application client
- possibly agree to the security-policies of Java,
- agree to the terms of licence and to select the OSCI directory.

Additionally, an information window is automatically displayed informing the user that in order to send a message the following processes are still necessary:

- establish a P.O.box
- change to the Outbox,
- sign the message if necessary and
- send the message.

This information window is important to make sure that new users who are not familiar with the application do not consider messages from the outbox as already sent.

In the next step the user is prompted to establish a P.O.box (by agreeing to the terms of use). After the P.O.box is established, the user has to enter the encryption PIN to access the user interface.

Scenario 2 – Application client is installed but not started

If the application client is already installed on the computer but not started, there is no need for an active internet connection, that is, the process is not interrupted if the download server is not available. The application opens automatically and a window for the P.O.box selection is displayed. The P.O.box opens after entering the PIN.

Scenario 3 – Application client is installed and started

If the application client is already installed and started

- a P.O.box can be open or
- a P.O.box cannot be open

If a P.O.box is open, the data can be transferred to the open P.O.box. If no P.O.box is open, a P.O.box has to be opened before further processing.

Data transfer to the P.O.box

As soon as the post box is opened, the following process starts in the background:

- the configuration file (`govello.txt`) is downloaded from the web server or file system and transferred to the application client,
- mandatory parameters of the configuration file are validated,
- afterwards the content data is transferred (XML,XSL,attachments)

During validation and transfer the message is already shown as a message that is built in the outbox.

The application client creates an OSCI message from the given information. If required, the message has to be signed with the according signature level.

After successful signing and sending, the message is moved to the "Sentbox" as usual.

It is possible to view the transmission protocol and the acknowledgement. The acknowledgement contains the time of entry on the addressee's OSCI-Manager.

Appendix 3 Directory structure

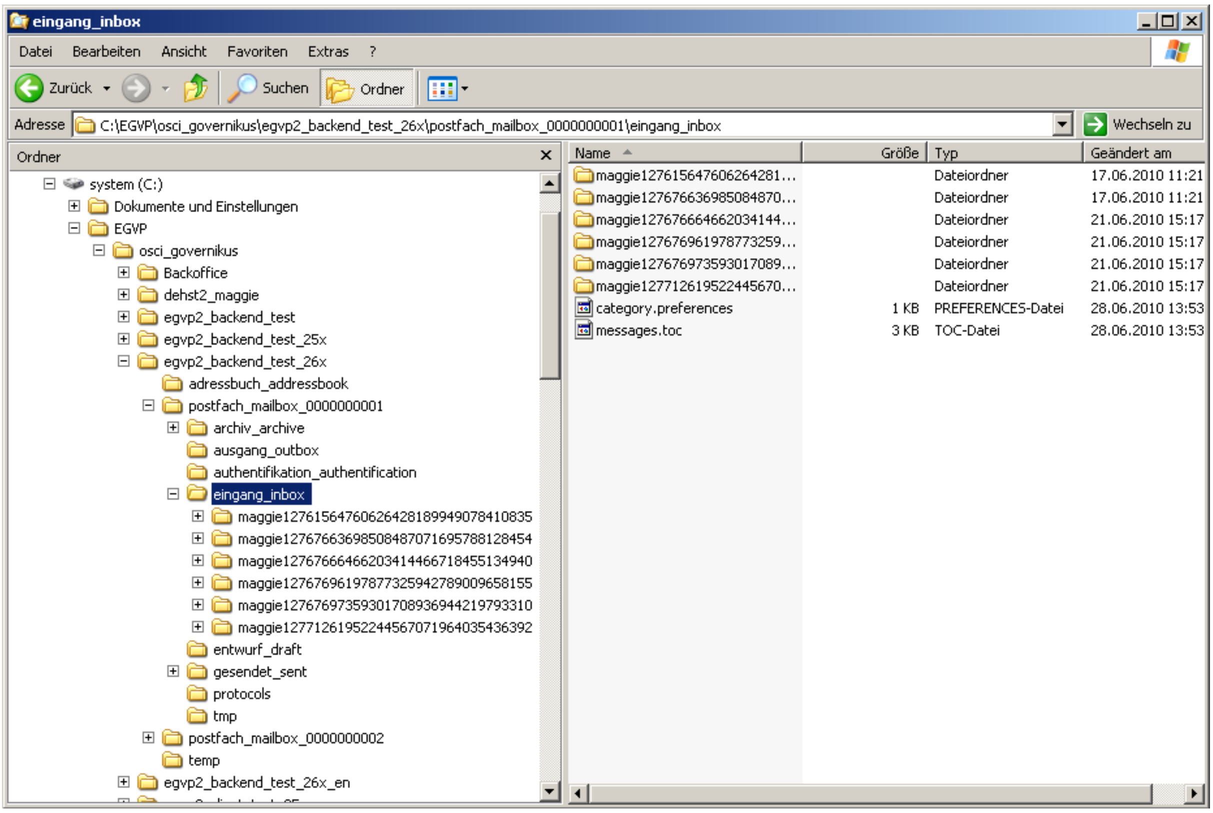
In the following the storage of the file system is explained. Variable parts in directory and file names being defined by the User or the system are marked with <>.

Attention:

Files and directories may not be changed. If files or directories are changed no guarantee for functionality can be granted.

<Root directory>

This directory, where message-folders are saved, is only defined during the first start of the application and cannot be changed offhand later. The name of the directory is saved once in the file"govello20.properties", which is stored in the computer's user directory.



Adresse: C:\EGVP\osci_governikus\egvp2_backend_test_26x\postfach_mailbox_0000000001\eingang_inbox

```
⊟ system (C:)
  ⊞ Dokumente und Einstellungen
  ⊟ EGVP
    ⊟ osci_governikus
      ⊞ Backoffice
      ⊞ dehst2_maggie
      ⊞ egvp2_backend_test
      ⊞ egvp2_backend_test_25x
      ⊟ egvp2_backend_test_26x
          adressbuch_addressbook
        ⊟ postfach_mailbox_0000000001
          ⊞ archiv_archive
            ausgang_outbox
            authentifikation_authentification
          ⊟ eingang_inbox
            ⊞ maggie12761564760626428189949078410835
            ⊞ maggie12767663698508487071695788128454
            ⊞ maggie12767666466203414466718455134940
            ⊞ maggie12767696197877325942789009658155
            ⊞ maggie12767697359301708936944219793310
            ⊞ maggie12771261952244567071964035436392
            entwurf_draft
          ⊞ gesendet_sent
            protocols
            tmp
        ⊞ postfach_mailbox_0000000002
          temp
      ⊞ egvp2_backend_test_26x_en
```

Name	Größe	Typ	Geändert am
maggie127615647606264281...		Dateiordner	17.06.2010 11:21
maggie127676636985084870...		Dateiordner	17.06.2010 11:21
maggie127676664662034144...		Dateiordner	21.06.2010 15:17
maggie127676961978773259...		Dateiordner	21.06.2010 15:17
maggie127676973593017089...		Dateiordner	21.06.2010 15:17
maggie127712619522445670...		Dateiordner	21.06.2010 15:17
category.preferences	1 KB	PREFERENCES-Datei	28.06.2010 13:53
messages.toc	3 KB	TOC-Datei	28.06.2010 13:53

Figure 6: detail of the file system

osci_governikus

The application saves all user information in a separated subfolder in this OSCI-directory.

<application_path>

Depending on the installed application client the directory name can vary, e.g. `egvp2_client`, `egvp2_backend`, `egvp2_slave`. This directory contains the following directories and files:

adressbuch_addressbook

The folder contains the local address book which is loaded after opening the P.O.box.

postfach_mailbox_<xxxxxxxxxxxxx>

It is possible to establish any number of P.O.boxes. A unique number is assigned to each P.O.box (e.g. 00000000001). All data from this directory is loaded after the P.O.box is opened.

archiv_archive

All received and sent messages are archived in this folder respectively saved a second time. All files in the archive folder are read only. Within the folder each message is stored in an own subfolder (message folder). The name corresponds to the unique ID given by the OSCI Manager.

ausgang_outbox

The outbox contains all messages to be sent. Within the directory each message is saved in a specific subfolder. The folder is named `message_<xxxxxxxxxx>`, with `xxxxxxxxxx` being a continuous numbering.

authentifikation_authentification

Every P.O.box has to authenticate against the registration server and stores the messages with certificates for encryption on the OSCI Manager. The folder contains information about the certificates.

eingang_inbox

The inbox contains all received messages. Within the folder the messages are stored in subfolders (message folders). The name corresponds to the unique ID given by the OSCI Manager.

entwurft_draft

In case a message has to be edited after creation it is saved temporarily in this folder to move it to the outbox after completion.

gesendet_sent

The folder contains all sent messages. Within the folder there are subfolders for each message. The folder-names correspond to the unique ID given by the OSCI Manager.

tmp

The folder contains files, which can be used by the application when needed.

favl

The file contains the local distribution list. It consists of a list of user IDs (Govello-ID) and/or distribution list IDs (DIST_ID). The complete address entry is included in the local address book or the replication database.

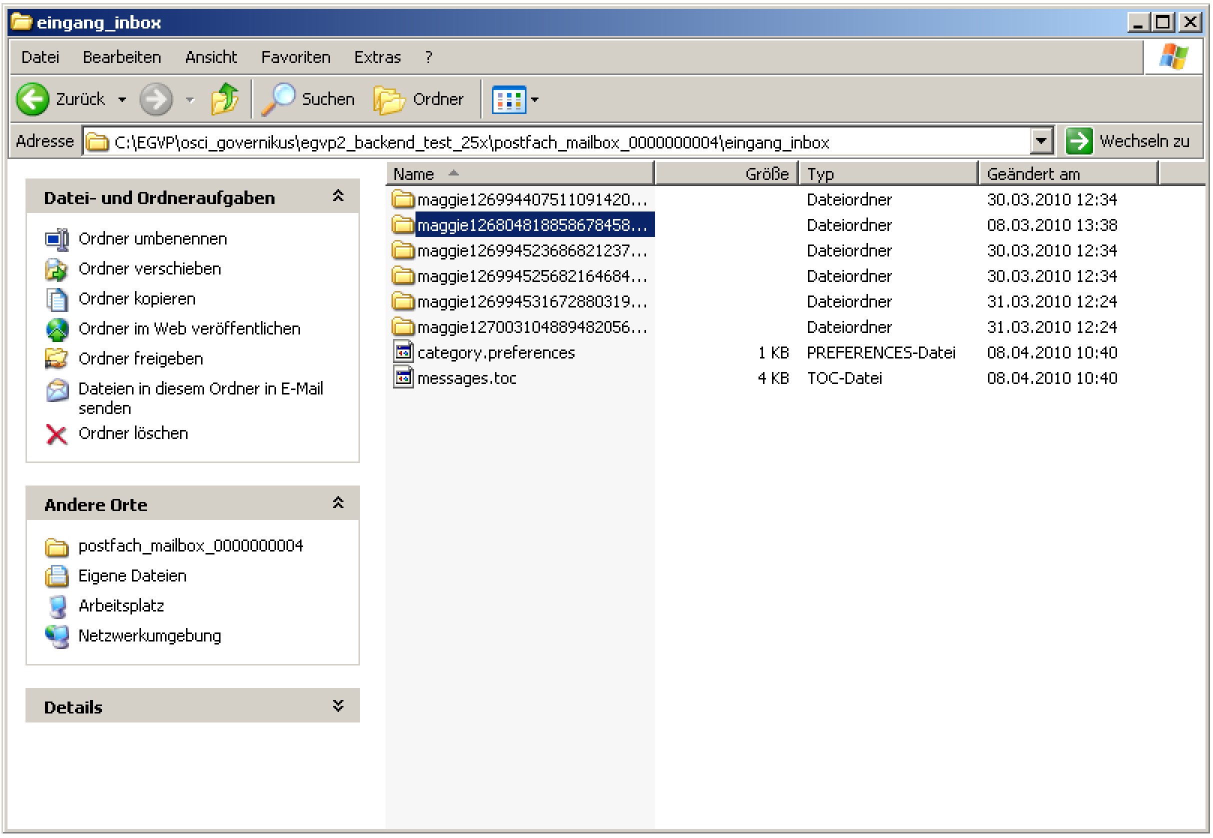
aidl

The file contains local distribution lists. Each distribution list consists of a collection of user-IDs (Govello-ID)

Note: Files which are not described explicitly are part of the system/directories and are necessary for the functionality of the application client.

Appendix 4 Message directory

The name of the message folder corresponds to the unique ID, which was given by the OSCI Manager. Excluded are the message folders in the outbox.



C:\EGVP\osci_governikus\egvp2_backend_test_25x\postfach_mailbox_0000000004\eingang_inbox

Name	Größe	Typ	Geändert am
maggie126994407511091420...		Dateiordner	30.03.2010 12:34
maggie126804818858678458...		Dateiordner	08.03.2010 13:38
maggie126994523686821237...		Dateiordner	30.03.2010 12:34
maggie126994525682164684...		Dateiordner	30.03.2010 12:34
maggie126994531672880319...		Dateiordner	31.03.2010 12:24
maggie127003104889482056...		Dateiordner	31.03.2010 12:24
category.preferences	1 KB	PREFERENCES-Datei	08.04.2010 10:40
messages.toc	4 KB	TOC-Datei	08.04.2010 10:40

Figure 7: message folder

Within the message folder the following files have to be saved:

File	Description
`oscicontentdata.osci`	The file contains the OSCI message.
`processcard.osci`	The file contains the process card signed by the OSCI Manager (only for sent messages in the folders gesendet_sent and archiv_archive)
`acknowledgement.html`	Acknowledgement as HTML-file, which is shown in the tab "acknowledgement" (only for sent messages in the folder gesendet_sent or. archiv_archive)
`acknowledgement.xml`	Acknowledgement as XML-file (only for sent messages in the folder gesendet_sent or. archiv_archive)
`businesscard.html`	Business card as HTML-file, which is shown in the tab "businesscard".

`inspectionsheet.html`	Inspection sheet as HTML-file, which is shown in the tab "inspection sheet" (only for received messages in the folders eingang_inbox and archiv_archive)
`message.html`	Message as HTML-file, which is shown in the tab "message"
`signedattachments.html`	Result of the verification of signed attachments as HTML-file, which is shown in the tab "signed attachments" (only for received messages in the folders eingang_inbox and archiv_archive)
`summarisedinspectionsheet.xml`	Results of the inspection sheet and of the signed attachments (if existing) as XML-file
`MsgProps.xml`	This file contains the meta data container.
`content.preferences`	This file is mandatory for working with a message.
`message.preferences`	The file provides information like subject, signer etc.
`signature.preferences`	The file contains all local signature times.
`additionals`	Subfolder containing the files for content data. nachricht.xml: covering note as XML-structure nachricht.xsl: style sheet for visualisation of the nachricht.xml visitenkarte.xml: business card as XML-structure visitenkarte.xsl: respective style sheet for the visualisation of the visitenkarte.xml herstellerinformation.xml: XML-structure for information of software's manufacturer, which is created together with the message additional: any further structured contents for content data.
`attachments`	Subfolder containing all files for attachments
`certificates`	Subfolder for all necessary certificates. `addressee.crt: Public key of addressee` `originator.crt`: Public key of sender `osci-manager.crt`: public key of the OSCI Manager `attribute`: Subfolder, which contains the PKCS#7 certificates. `authors`: Subfolder, which contains the certificates of the signer.

	pkcs7: Subfolder, which contains the the certificates of the signer PKCS#7. readers: Subfolder, which contains the certificates of the reader.
project	Subfolder containing a printable summary of the message for filing.

Table 1: Files in the message folder

4.1 Appendix 5 Error messages - Exchange of data via file system

In the following possible error messages are listed. All messages start with the ID of the affected message followed by "The conversion into an OSCI message failed". After a detailed error message as seen in the listing below the information, the user is informed that the message has been moved to "DefectiveMessage".

The error message ends with the advice "Please contact the manufacturer of your 3rd party application".

Description	**Error message**
The addressee's ID, which is specified in the configuration file, is not known by the address book/Registration server	Der Empfänger wurde nicht im Adressbuch gefunden bzw. wurde falsch angegeben. Ggf. sollten Sie das Adressbuch aktualisieren und den gesamten Vorgang nochmals durchführen. (German text only due to technical reasons)
The parameter "ID" is not specified in the configuration file.	The indication of the parameter –ID– is missing.
The conversion of the XML- and XSLT-file to an HTML-file failed.	The conversion of the XML and XSLT files into HTML failed.
The application client is not able to find the XML-file, which is specified in the configuration file.	The indication of the parameter –message– is not accurate.
The application client is not able to find the XSLT-file, which is specified in the configuration file.	The indication of the parameter –stylesheet– is not accurate.
The parameter "key" is specified incorrectly in the configuration file.	The indication of the parameter –key– is not accurate.
The parameter "key" is not specified in the configuration file.	The indication of the parameter –key– is missing.
The parameter "signature level" is specified incorrectly in the configuration file.	The indication of the parameter –siglevel– is not accurate.
The parameter in the "signature level" is	The indication of the parameter –siglevel– is

not specified in the configuration file.	missing.
The parameter "type" is not specified in the configuration file.	The indication of the parameter –type– is not correct.
The parameter in "type" is specified incorrectly in the configuration file.	The parameter –type- is not accurate.
The maximum number of attachments has exceeded.	The number of allowed attachments has exceeded.
The maximum overall size of attachments is exceeded.	The limited message size has exceeded.
The configuration file is missing in the message folder.	The file –govello.txt- was not found.
The configuration file is unreadable.	The file –govello.txt- is invalid.
A file could not be accessed.	Part of the message could not be loaded.
While creating the message an error occurred during the conversion of the data to an OSCI message.	The OSCI message could not be created.

Table 2: Error messages during exchange via file system

4.2 Appendix 6 Error messages - Dynamic transfer of data

In the following, possible error messages of the interface are listed. All error messages start with the ID of the message, followed by "During the conversion into an OSCI message an error occurred." After a detailed error message as seen in the listing below there are two different variants:

- Variant 1: The message is moved to the outbox. You can either try to repeat the process from the context menu or delete the message manually from the outbox.
- Variant 2: The message is moved to the outbox and has to be deleted manually.

The error message ends with the advice "Please contact the manufacturer of your 3rd party application"

Var.	Description	Error message
1	The addressee's ID which is specified in the configuration file is not known by the address book/Registration server	Der Empfänger wurde im Adressbuch nicht gefunden. Ggf. sollten Sie das Adressbuch aktualisieren. (German text only due to technical reasons)
2	The parameter "ID" is not specified in the configuration file.	The parameter –id- is missing.
2	The conversion of the XML- and XSLT-file to an HTML-file failed.	The conversion of the XML and XSLT files into HTML failed.

1	The application client is not able to find the XML-file, which is specified in the configuration file.	Loading the XML file <file name> failed.
1	The application client is not able to find the XSLT-file, which is specified in the configuration file.	Loading the XSLT file <file name> failed.
2	The parameter "key" is specified incorrectly in the configuration file.	The parameter –key- is not correct.
2	The parameter "key" is not specified in the configuration file.	The parameter –key- is missing.
2	The parameter "signature level" is specified incorrectly in the configuration file.	The parameter –signlevel- is not correct.
2	The parameter in the "signature level" is not specified in the configuration file.	The parameter -signlevel- is missing.
2	The parameter "type" is not specified in the configuration file.	The parameter –type- is missing.
2	The parameter in the "type" is specified incorrectly in the configuration file.	The parameter –type- is not correct.
1	The configuration file could not be loaded.	Loading the control file <file name> failed.
1	The attachments specified in the configuration file could not be loaded.	Loading the attached file <file name> failed.

Table 3 Error messages during dynamic data transfer