STORM Manual

Release 2024.3.1

OTRS AG

Apr 22, 2024

CONTENTS

CHAPTER

ONE

DARK THEME

STORM introduces an own dark theme for the login pages and for the agent interface as well as a new dark skin for the administrator interface. The dark theme and the dark skin are enabled by default.

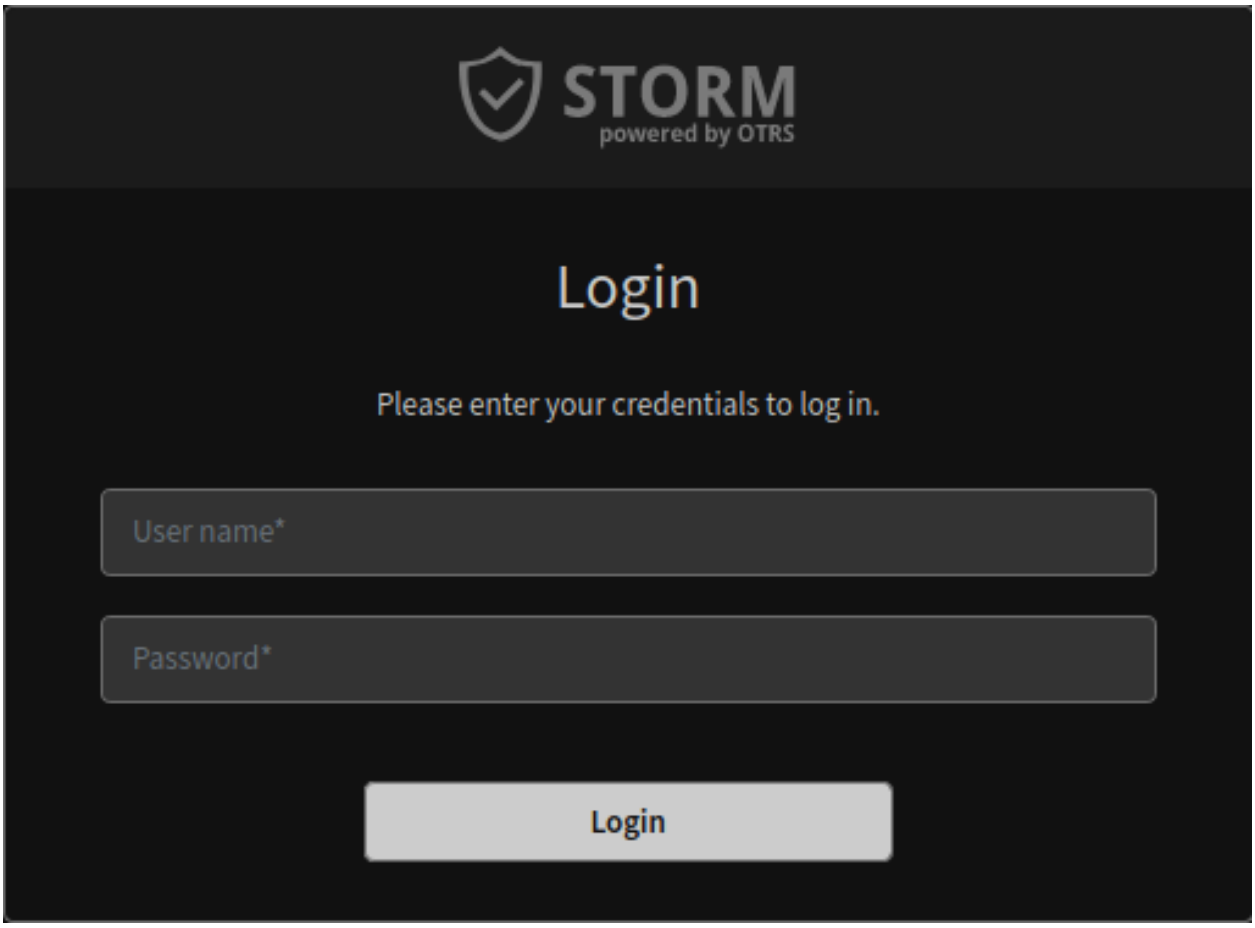

Fig. 1: Login Box With Dark Theme

The agents can restore the default OTRS theme and they can select any other theme, that is familiar from the OTRS framework.

See also:

Please refer to the user manual how to change the theme.

The administrators can change the skin in the agent preferences.

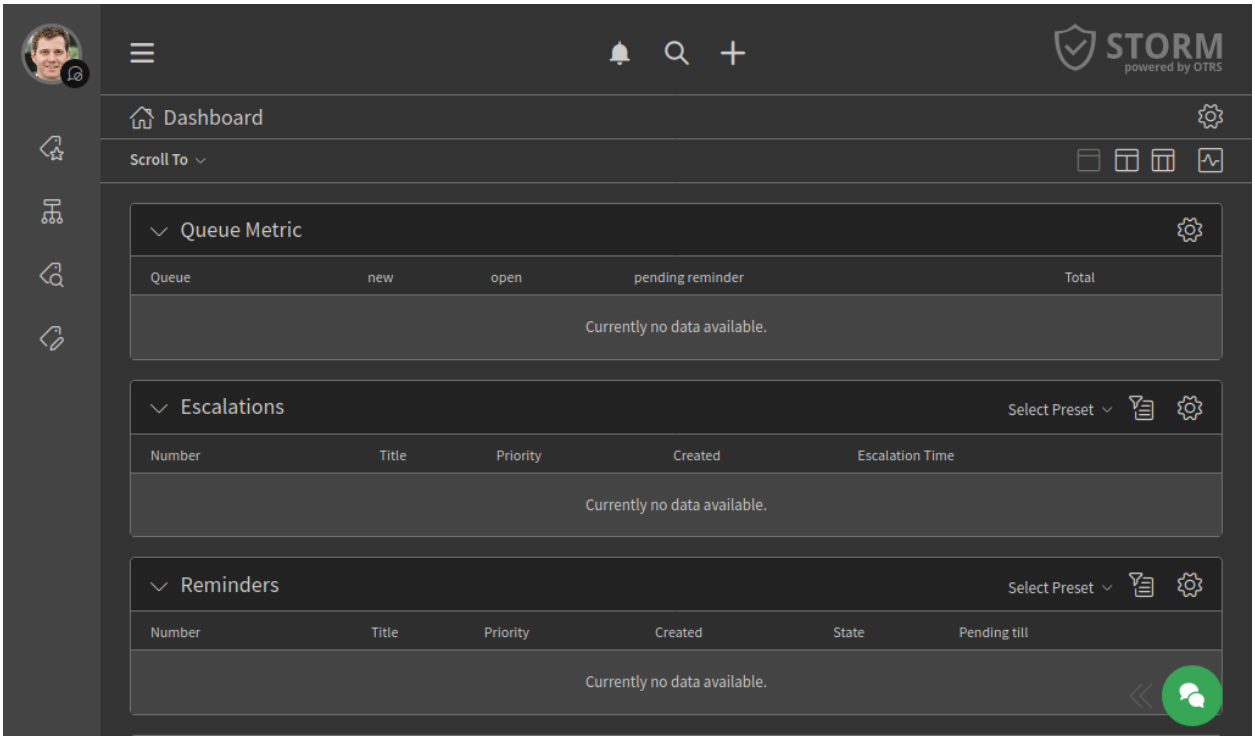

Fig. 2: Agent Interface With Dark Theme

To change the skin:

1. Go to the *Agents* module in the administrator interface.
2. Select the agent from the list of agents.
3. Click on the *Edit personal preferences for this agent* button in the left sidebar.
4. Select the *Miscellaneous* group.
5. Change the skin in the *Administrator Interface Skin* section.

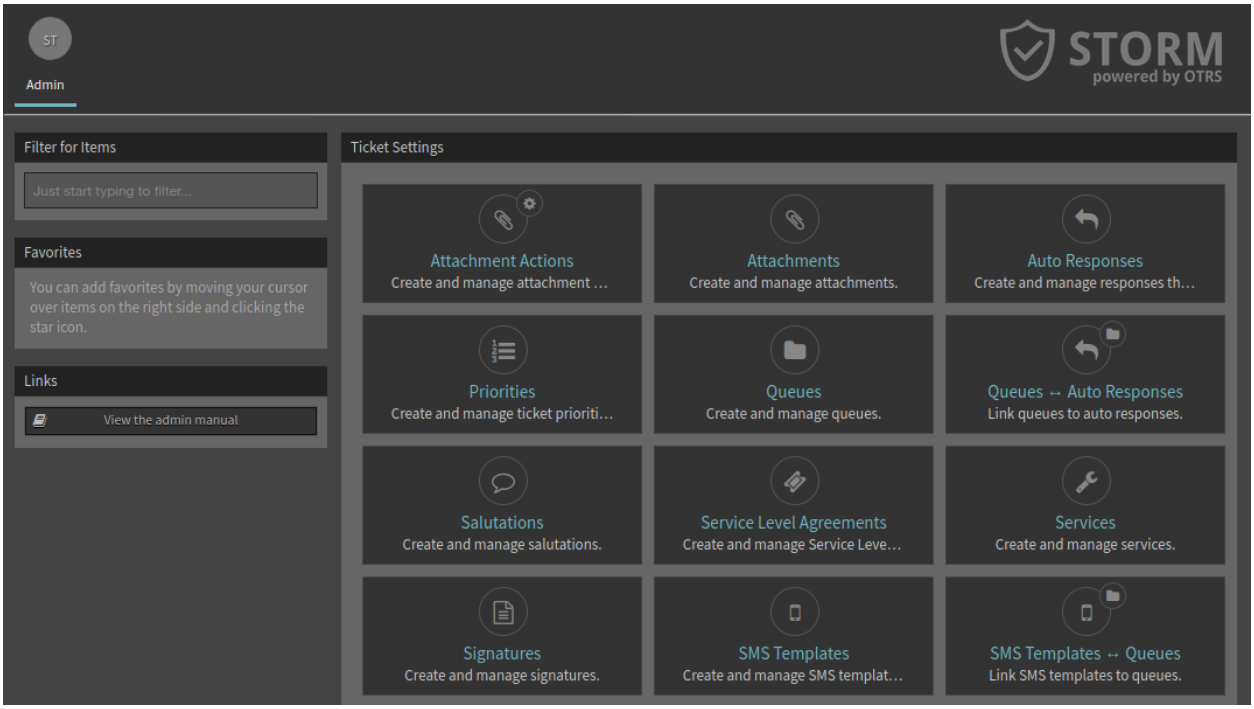

Fig. 3: Administrator Interface With Dark Skin

CHAPTER

TWO

COMMUNICATION RESTRICTION

Outgoing communication from the application is restricted by default. The restrictions can be removed via the system configuration. The restrictions limit the following features:

News Widget
The default configuration of the *News* widget would need a web service call to cloud.otrs.com and is therefore deactivated by default.

Package Manager
The package manager has two ways to operate. The administrator can upload and install the package manually or an online repository can be used. This repository is deactivated by default. Also the verification mechanism for packages is deactivated. So *OTRSVerify* will not work and a warning might occur in the web interface.

Cloud Services
Automatic cloud service connection to the OTRS Group are deactivated by default. This will restrict the usage of SMS, automatic license check and registration update. To perform the needed license check an administrator has to run it manually via the *STORM Management Module*.

Disabled Gravatar
Gravatar is a third party service to include user avatars as profile image in the user cards and in the communication stream. The hashed email address of the particular user would be transferred to an external service and is therefore deactivated by default. The initials of the users will be displayed instead.

CHAPTER

THREE

STORM MANAGEMENT MODULE

There is a change in the system configuration that restricts the normal communication between the STORM instance and the OTRS Group.

Due to the communication restriction, it is not possible to sent the registration information on a regular basis. In the OTRS framework this is handled by the daemon, but in STORM this is not done automatically. However, there is a separate module *STORM* in the *OTRS Group Services* group of the administrator interface. Use this screen to send registration updates and contract status checks manually, to fit the conditions of your security environment.

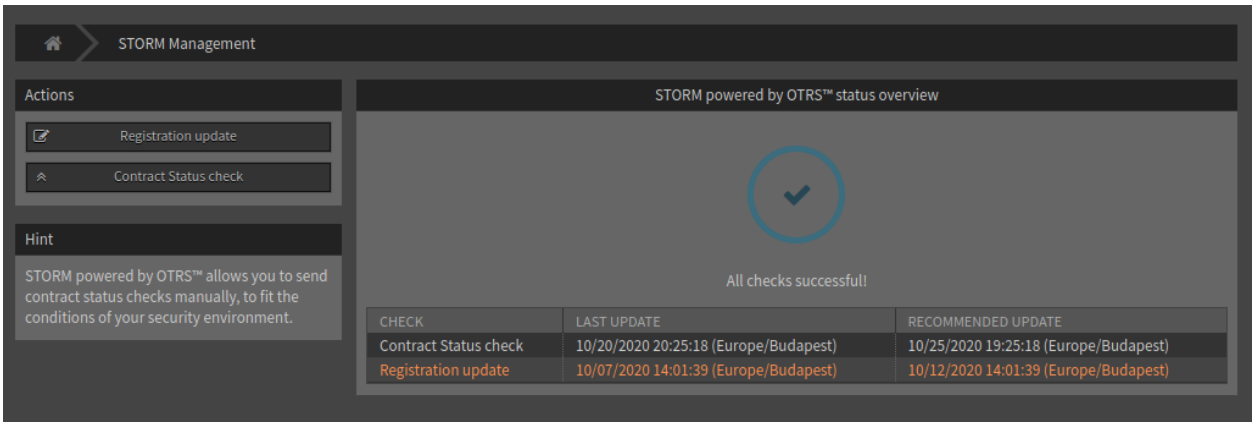

Fig. 1: STORM Management Screen

The preview of the data to be sent can be reviewed before sending it. This method ensures that no sensitive data will be send to OTRS Group.

To send a registration update:

1. Click on the *Registration Update* button in the left sidebar.
2. Review the system registration data that going to be sent to the OTRS Group.
3. Make sure, that the communication is not blocked to the OTRS Group.
4. Click on the *Transmit* button.

To check the contract status:

1. Click on the *Contract Status Check* button in the left sidebar.
2. Review the contract status data that going to be sent to the OTRS Group.
3. Make sure, that the communication is not blocked to the OTRS Group.
4. Click on the *Transmit* button.

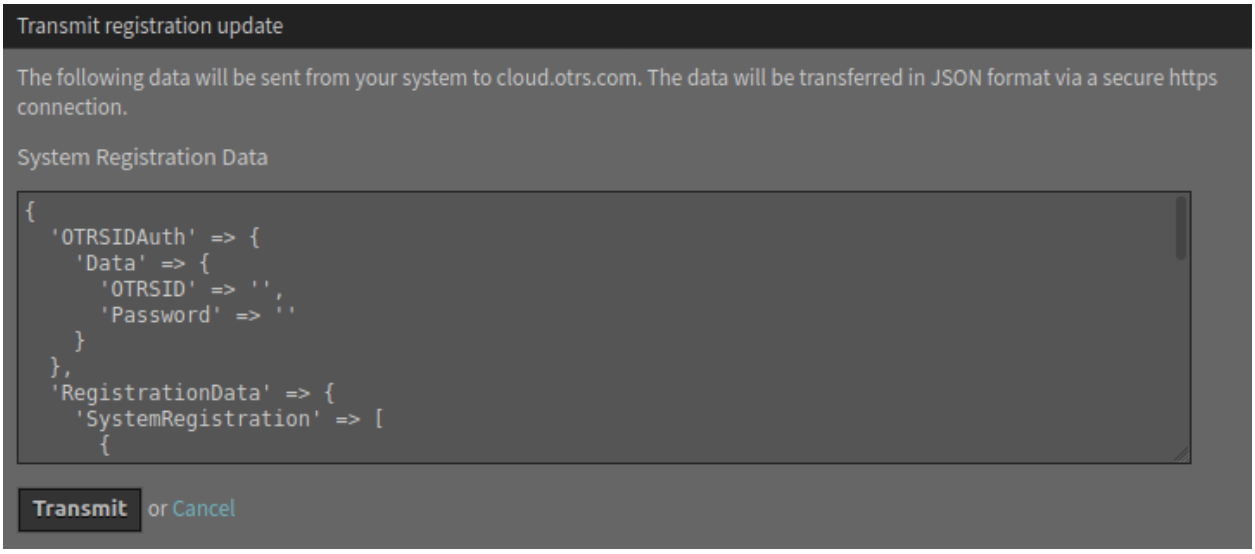

Fig. 2: Registration Update Screen

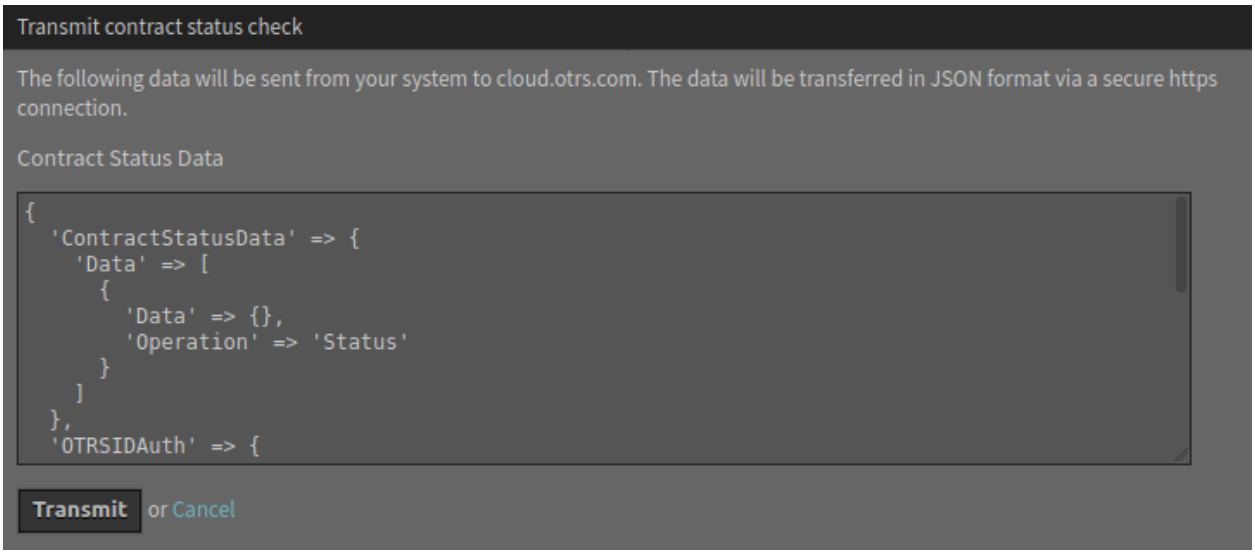

Fig. 3: Contract Status Check Screen

CHAPTER

FOUR

ARTICLE SEEN HISTORY

This feature is used to ensure the auditability of the system for critical information. This function allows articles to be displayed in the history in such a way that it is visible who has read the article.

4.1 Requirements

The system configuration setting `UserArticleSeenHistory` needs to be enabled.

4.2 Usage

The feature adds an entry to the history, when an agent reads an article.

To see the article seen history:

1. Open a ticket in the ticket detail view.
2. Select *View History* in the menu *Actions*.

The entries for the notifications about that a person has read the article are shown in the history.

Reading means in this case, that the agent has opened the article detail view. In this case the `IsSeen` flag is set to `1` and in the ticket history an entry is created with the information which person has read the article.

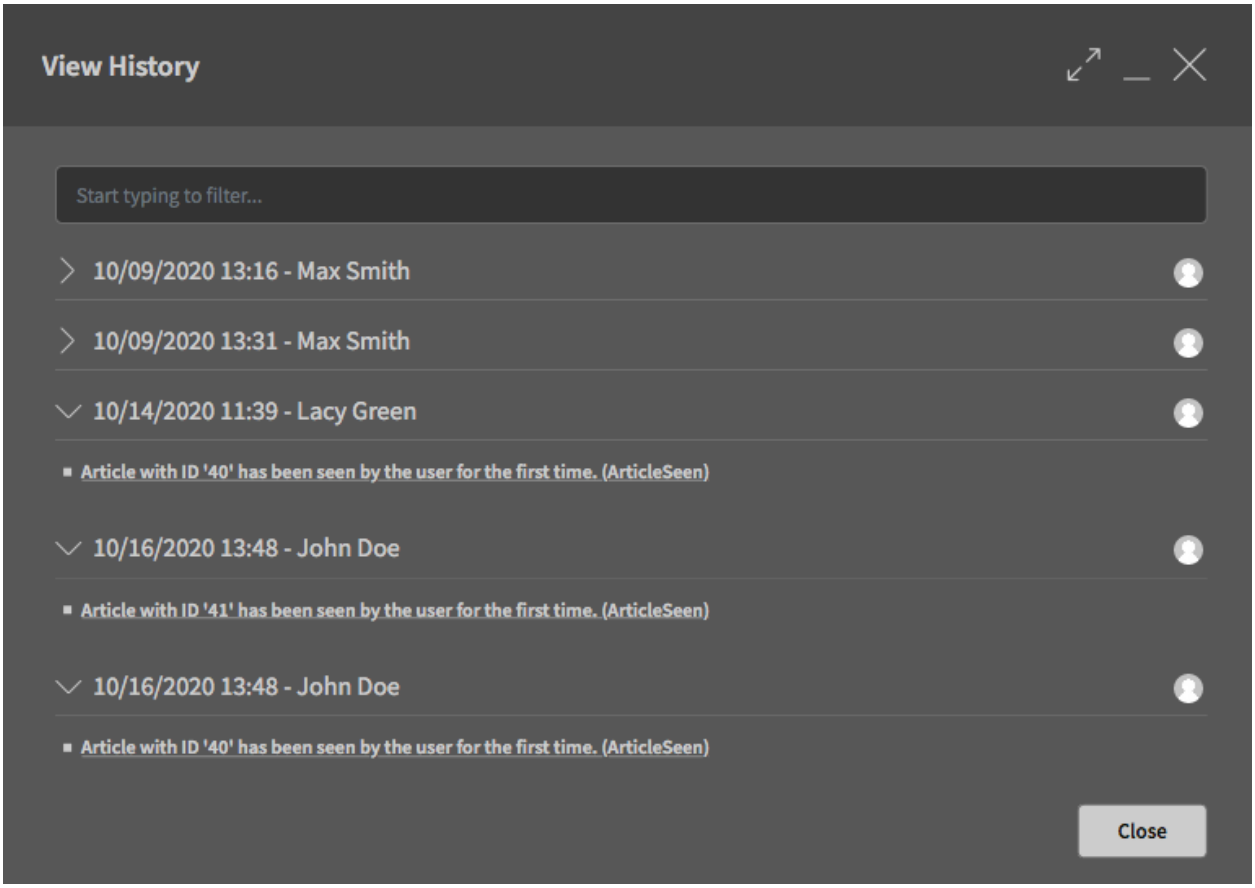

Fig. 1: Example Article Seen History

CHAPTER

FIVE

ARTICLE RAW SOURCE FOR WEB SERVICE

If articles in web services are sent with only their body, they usually contain too little detail for deeper and more secure analysis. Security analysts benefit when an email article with all its raw data, including all headers, is sent to remote systems for deeper analysis.

For this purpose STORM has a function to send the raw email article data to a remote system with the generic interface operations and invokers.

If OTRS works as provider, this data is automatically sent when you set up the web service. This is only possible for the operation `TicketGet`.

If OTRS works as requester, the field needs to be selected if setting up the invoker settings by choosing the value `ArticlePlain`.

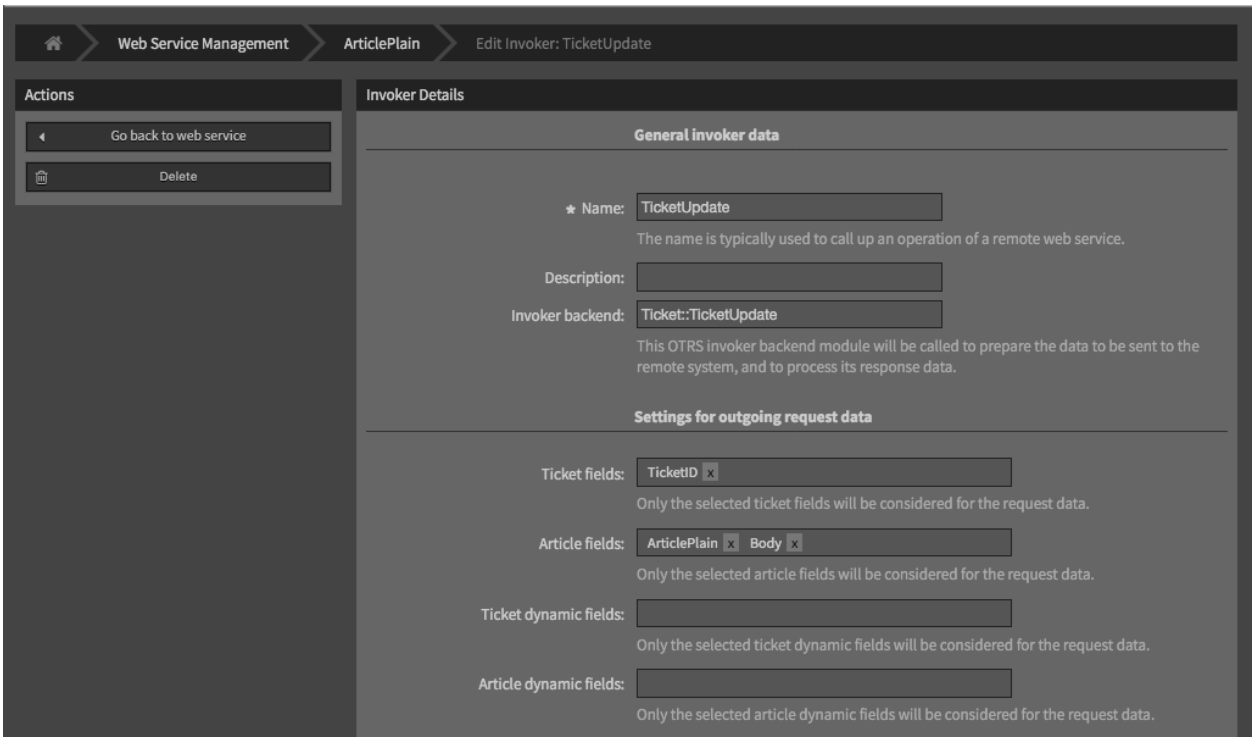

Fig. 1: Web Services Invoker Settings

The article plain field can be used for the invokers `TicketCreate` and `TicketUpdate`.

The article raw source data can be seen in the web service communication payload. For testing purposes, this information can be also found in the debugger.

CHAPTER

SIX

ATTACHMENT ACTIONS

This feature enables the execution of different custom actions over ticket attachments. These actions could come from modules such as the `ScanWithVirusTotal` module or from web services that administrators can define in order to send attachment information to a third party system for analysis, process, count, etc.

In order to send the attachment information to a third party server it might be needed to be extracted or transformed from the OTRS format to a format that the other system can understand. Also the response from the other system needs to be converted to a special format in order to be processed and recorded by the attachment actions. This data format change or transformation can be done by using the mapping modules in OTRS generic interface, especially the XSLT mapping module should be capable to accomplish this task.

6.1 Setup VirusTotal Module

The system already comes with a module to send attachments to be checked by *VirusTotal* via upload of the attachment. The attachment action associated to this module is not enabled by default.

To activate the virus scan module:

1. Go to the VirusTotal website and create an account.
2. Find and copy the API key provided by VirusTotal to use their web services.
3. Add the API key to the `AttachmentAction::ScanWithVirusTotal::APIKey` setting.
4. Enable the VirusTotal attachment action in the *Attachment Action Management* screen (see below).

Note: More module based attachment actions might be added later to STORM.

6.2 Create Web Services

Attachment actions can also use web services instead of predefined modules. This let the administrator to integrate their actions with remote servers as needed using XSLT mappings to transform data outbound and inbound.

Attachment actions should use the invoker `Ticket::AttachmentAction` as it prevents to send other attachments in the request and it also knows how to handle the results. This invoker comes with STORM.

After the inbound mapping the invoker should provide the key `<AttachmentActionResult>` with the following sub keys:

<Status>
A number from 1 to 6. The list of status codes and proposed usage are the following:

- `1` (Alert): Currently not in use (color purple).
- `2` (Critical): Used for internal server errors (color purple).
- `3` (Error): Execution errors (color red).
- `4` (Warning): Execution was correct but external errors reported (color orange).
- `5` (Notice): Execution was correct but results are not present or represent minor issues (color yellow).
- `6` (Info): Everything is fine (color green).

<Result>
A string to be displayed as a tool tip.

<Details>
Full result details in plain text format.

The web services can be created in the *Web Services* module of the administrator interface. The usage of this management screen is identical with the usage of the web service management screen of the OTRS framework.

Here is an example for XSLT mapping:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <xsl:copy>
            <RootElement>
                <AttachmentActionResult>
                    <Status>5</Status>
                    <Result>Web service sampple result</Result>
                    <Details>This is an example</Details>
                </AttachmentActionResult>\r\n
            </RootElement>
        </xsl:copy>
    </xsl:template>
</xsl:stylesheet>"
```

6.3 Manage Attachment Actions

After the web service was created by the administrator, it is necessary to create a new attachment action where the web service name has to be set and the invoker from the drop-down list has to be selected. There is a new module to manage the attachment actions. The attachment actions management screen is available in the *Attachment Actions* module of the *Ticket Settings* group in the administrator interface.

To add a web service as attachment action:

1. Click on the *Add Attachment Action* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

It is possible to create attachment actions for modules or web services. Two modules `ScanWithVirusTotal` and `ReportWithVirusTotal` are shipped with STORM, while new web services can be defined by the administrators.

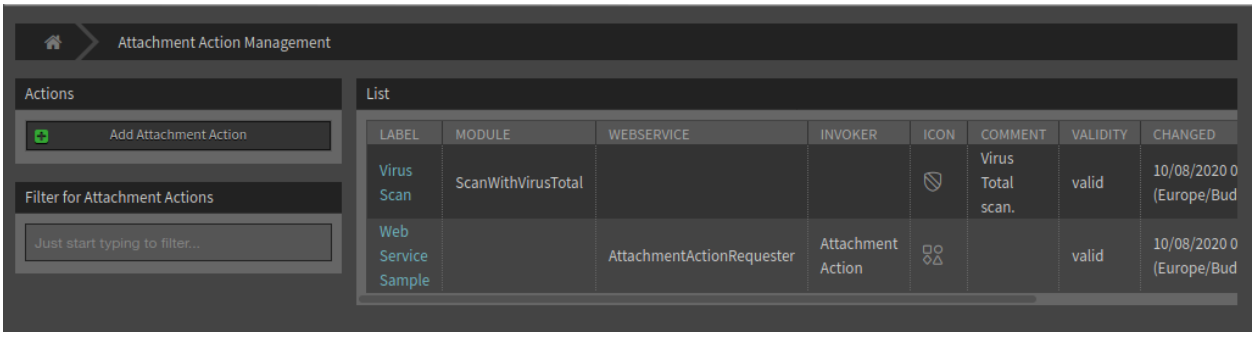

Fig. 1: Attachment Action Management Screen

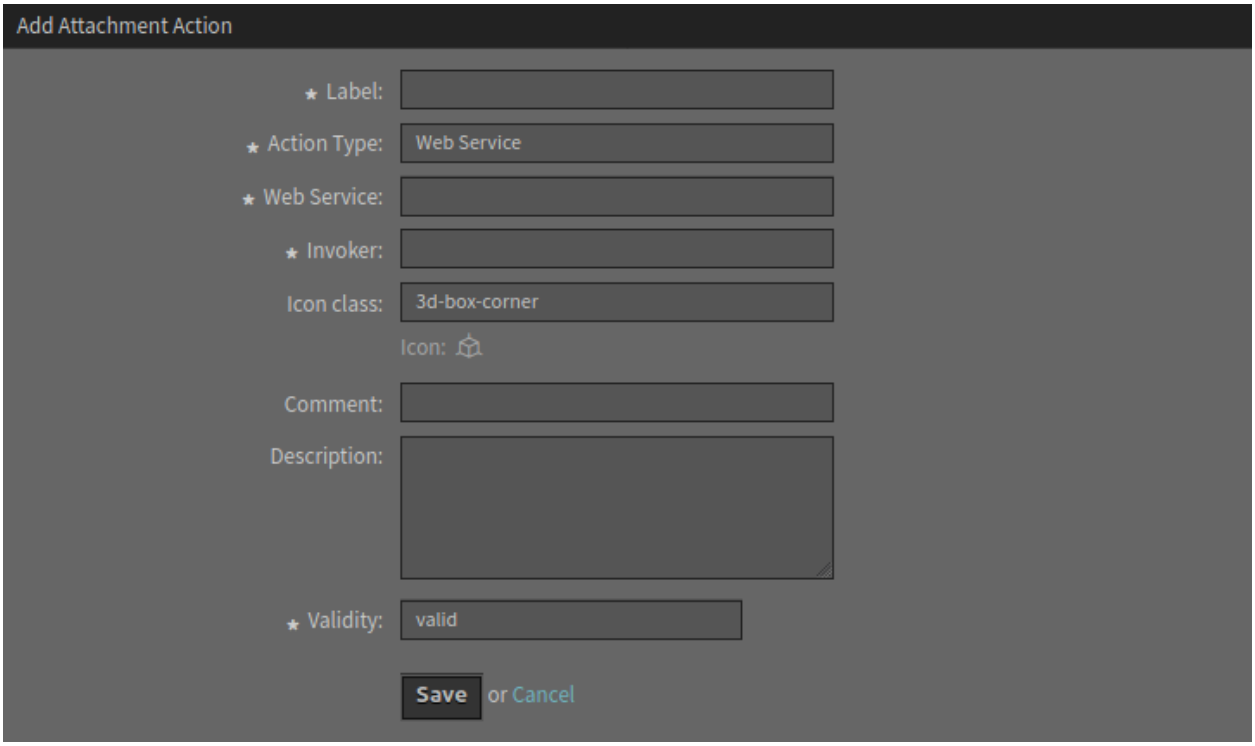

Fig. 2: Add Attachment Action Screen

Warning: Attachment actions can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit an attachment action:

1. Click on an attachment action in the list of attachment actions.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

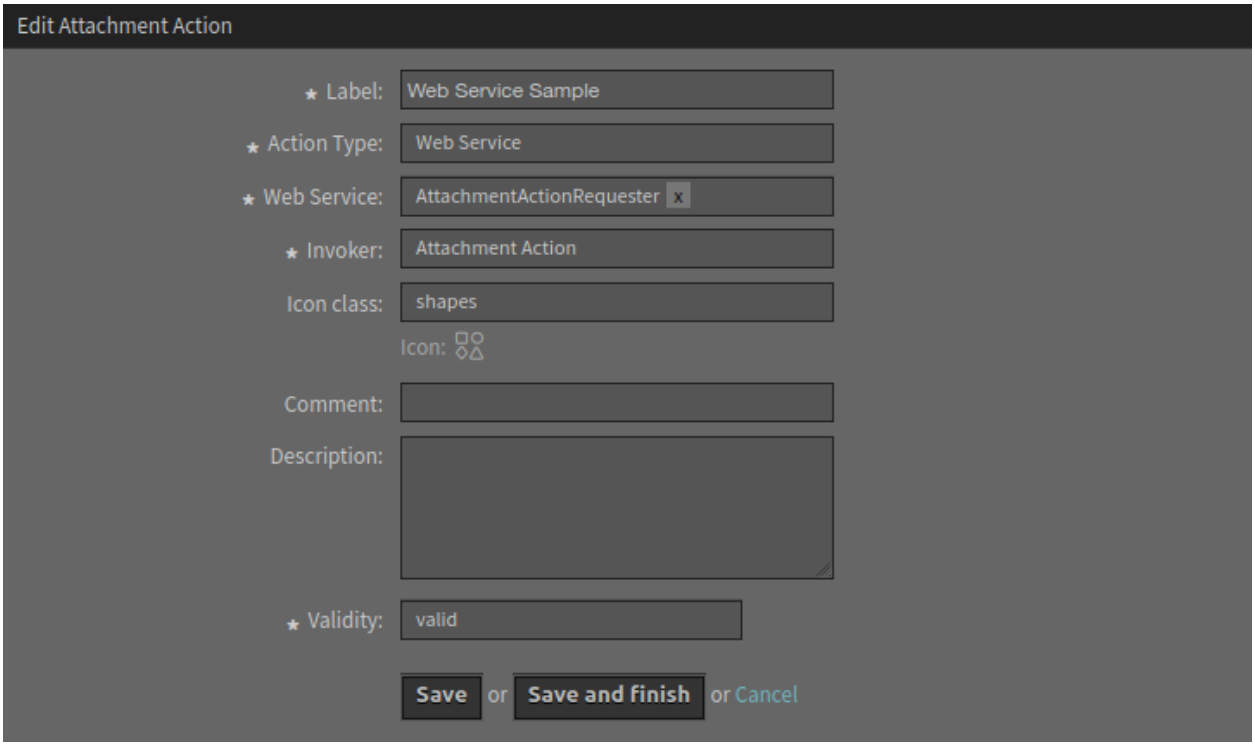

Fig. 3: Edit Attachment Action Screen

6.4 Usage

The attachment actions can be used in any attachment widget of the detail views.

To use the attachment actions:

1. Create a new ticket.
2. Fill in the required fields.
3. Add some attachments.
4. Go to the ticket detail view and find the *Attachments* widget.
5. Any attachment action has an own column in the *Attachments* widget.

The icons displayed in the widget is the same as set up for the action in the administrator interface. The color of the icons has been explained above.

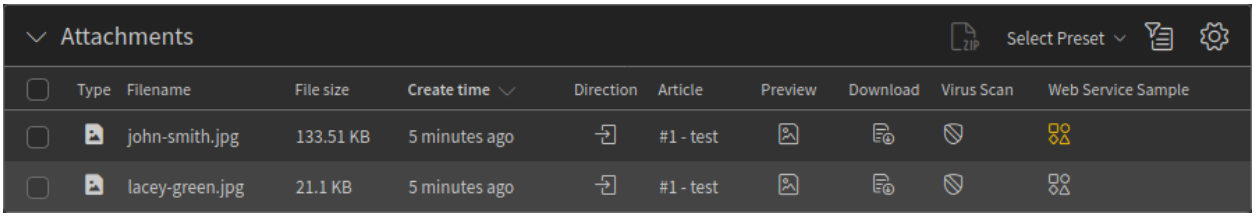

Fig. 4: Attachments Widget

Note: A column will be added for each attachment action. Try to define as many attachment actions as really needed, otherwise the widget might not fit in small screens.

6.5 Attachment Actions for VirusTotal

STORM provides two built-in attachment actions using the web service API from virustotal.com. These actions and their results are shown as separate columns in the *Attachments* widget.

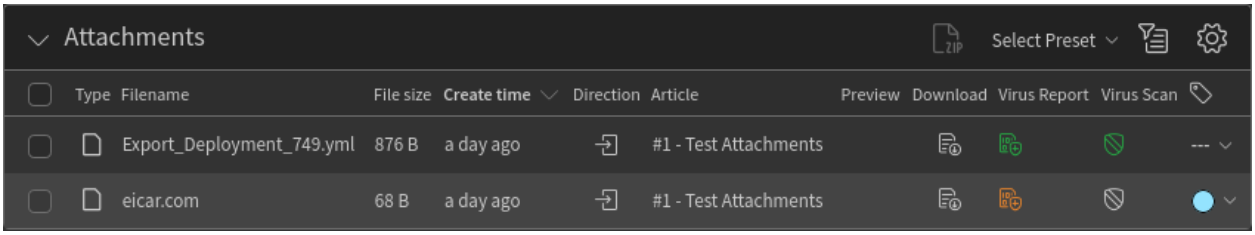

Fig. 5: Attachments Widget

The icons within the columns are used to perform the attachment action and to display the results of the analysis.

6.5.1 Virus Scan

The *Virus Scan* column is used to send an attachment to VirusTotal for virus scanning. In this case the file is sent to VirusTotal and VirusTotal returns a result after analysis whether this file contains a virus.

The results of this analysis are represented by the color of the icon. The colors have the following meaning:

- Green = No virus found
- Yellow = The file has been analyzed, but there are no results yet
- Orange = The file contains a virus
- Purple = Server error
- Gray = The file has not been analyzed yet

6.5.2 Virus Report

In some cases, it may be necessary, instead of sending an attachment directly to remote virus services, to send a hash of the data of this attachment, which will be used as an identifier by VirusTotal.

For this purpose STORM has a function that allows to send a hash instead of the attachment itself. This function is represented with a separate icon in the *Virus Report* column. If an agent clicks on this icon, only the data hash of this file will be sent to VirusTotal instead of the file itself.

VirusTotal searches this hash in their records and returns the information whether this file contains a virus. The results of this analysis are represented by the color of the icon. The colors have the following meaning:

- Green = No virus found
- Yellow = The file has been analyzed, but there are no results yet
- Orange = The file contains a virus
- Red = The hash was sent but VirusTotal has no file to compare with
- Purple = Server error
- Gray = The file has not been sent

CHAPTER

SEVEN

ATTACHMENT DOWNLOAD LOG

If attachments contain sensitive data and information, security managers benefit from logging of attachment downloads. More specifically, they can check who downloaded an attachment and related details. This allows them to pass security audits without stress. With using STORM it is possible to display in the system log the users who have downloaded an attachment.

This feature does not have any user interface, it only logs the activities in the system log. However, the *System Log* module of the *Administration* group in the administrator interface can be used to review the log entries.

7.1 Setup

The following system configuration settings have to be changed to enable the feature.

- `MinimumLogLevel` → *info*
- `UserAttachmentDownloadLog` → enabled

The following system configuration setting defines an optional prefix for the log entries. This makes it easier to parse the log file.

- `UserAttachmentDownloadLog::MessagePrefix`

7.2 Usage

As an agent, go to the ticket detail view of a ticket, which has some attachments and download any attachment. As an administrator, check the system log.

The attachment downloads data are displayed as log entries. If the prefix for attachment download is defined then the entries contain this prefix.

```
Thu Oct 22 15:16:52 2020 (Europe/Berlin)   info   WebApp-10   ATTACHMENT - Download␣
↪of 'Inquiry.pdf' (ticket '2020102210000033') by 'John Smith'.
```

Note: If the *Dynamic Field Attachment* feature is installed, the downloads of the attachments in the dynamic fields are also logged in the system log.

CHAPTER

EIGHT

DOCUMENT SEARCH ARTICLE META FILTERS

With the article meta filters you can define a configuration with regular expression of search criteria you would like to search for inside an article. The feature can provide links that uses these search criteria you searched for in an article. This is similar to the CVE numbers meta filter built in the OTRS framework.

The idea of this feature is to provide a very similar feature as already present in the OTRS framework, but instead of search based on some criteria on the internet or display something from the internet we want to have this meta filter make use the document search engine to search for anything you would like to search in an article and inside other objects of OTRS like tickets, knowledge base articles, appointments or any other business objects.

By default, there are some article meta filters shipped with STORM. If you search for host names, servers or IP addresses, it creates buttons with links to the document search.

8.1 Setup

The feature can be enabled with the `AgentFrontend::TicketDetailView::ArticleMeta` setting. This setting is required for the meta filters built in the OTRS framework, but this is also required to the document search article meta filter.

There are three examples in the `AgentFrontend::TicketDetailView::ArticleMetaFilters::DocumentSearch` setting, but all of them are inactive by default. To activate any of them, just change the value of the `Active` key to *1*.

The first example will search for host names, the second example will search for servers, and the third example will search for IP addresses. You can see what regular expressions are defined in the `RegExp` array.

There is an other setting `AgentFrontend::TicketDetailView::ArticleMetaFilters::DocumentSearch###000` where the administrators can define custom meta filters.

Note: It is not recommended to change or extend the examples, because the built in examples can be changed in the future. Please use the custom setting to define the own meta filters.

The preview feature requires an additional setting. The fully qualified domain name (FQDN) of the STORM instance have to be added to the `frame-src` key of the `WebApp::Server::AdditionalOrigins` setting. Otherwise the preview feature will not work.

8.2 Usage

This example will show how to use this feature to search for IP addresses. For this, one of the built in examples is used. We assumed, that this example meta filter is activated as described above.

To see all article possibilities of the feature, appointments, knowledge base articles and tickets are needed which have an IP address (*192.168.0.1* and *255.255.255.0*) in its text fields. For this example:

1. Create an appointment with an IP address in the description.
2. Create a knowledge base article with the same IP address in the *Symptom* or *Problem* fields.
3. Create a couple of tickets with articles that contain the same IP address.

To search for IP addresses:

1. Create a new ticket.
2. Fill in the required fields.
3. Enter the following text in the body: *Your IP address is 192.168.0.1 and your subnet mask is 255.255.255.0*.
4. Go to the ticket detail view of the newly created ticket.
5. Expand the first article in the *Communication Stream* widget to see the buttons below the article.

The engine will search for all possible IP addresses in the article as configured by the regular expression.

The buttons point to the search results of a document search. This should be returned the same search results when an agent starts a search for the given IP addresses. The text for the buttons (*IP Address* in this example) comes from the `Label` key of the underlying system configuration setting.

If the agents hover the mouse over a button, they will get a preview of the search results screen. Clicking on the buttons will open the search results screen.

This feature works for all articles of a ticket.

CHAPTER

NINE

WEB SERVICE ARTICLE META FILTERS

Security analysts want to find relevant IPs and other data from messages in existing records, so that they can save time for investigating by using the meta data collector web service interfaces, rather than wasting time for manually searching for existing matches of received IPs or other data, whenever messages contain data for investigation.

The principle of this feature is the same as other article meta filters. Some regular expressions can be defined to call a web service invoker. Depending on the web service and on the external server what the web service calls, the response will contain a list of results. The list of results should come in a particular format so that OTRS can understand. It is highly recommended to use an XSLT mapping for this invoker, so it can convert the results got from the external provider to what we can understand in OTRS.

9.1 Setup

The feature can be enabled with the `AgentFrontend::TicketDetailView::ArticleMeta` setting. This setting is required for the meta filters built in the OTRS framework, but this is also required to the web service article meta filter.

There are some examples in the `AgentFrontend::TicketDetailView::ArticleMetaFilters::WebService###0` setting, but all of them are inactive by default. To activate any of them, just change the value of the `Active` key to *1*.

- The first meta filter is just an example how to search for host names using Google.
- The second meta filter is another example how to search for servers using Google.
- The third meta filter is a more complex example how to search for IP addresses using a "who is" service.
- The fourth meta filter can bring information of IP addresses.
- The fifth meta filter can bring information of vulnerability issues.

It may be necessary to set the correct values for the `WebService`, `Invoker` and `Payload` keys in the first three examples to match the current system. The fourth and fifth meta filters are real world examples, they should work without any change after activation.

You can see what regular expressions are defined in the `RegExp` array.

There is an other setting `AgentFrontend::TicketDetailView::ArticleMetaFilters::WebService###0002-Cu` where the administrators can define custom meta filters.

Note: It is not recommended to change or extend the examples, because the built in examples can be changed in the future. Use the custom setting to define the own meta filters or copy the content from the

example and extend it there.

In the configuration is needed to specify which web service and which invoker is going to be called. The remote server should return a list of elements. This list will be displayed in a popup window in the article, if the article has some keywords that matches with the configured regular expression.

The `Payload` is the information that OTRS sends to the remote server. This information is specified by the remote web service provider and it could contain static data or the match or matching groups specified in the `RegExp` array. The `Payload` can contain references to the `TicketID`, `ArticleID` and `TicketNumber` through the OTRS smart tags `<OTRS_TICKET_TicketID>`, `<OTRS_TICKET_ArticleID>` and `<OTRS_TICKET_TicketNumber>`.

Example:

```
Payload:
  # ...
  TicketID: <OTRS_TICKET_TicketID>
  ArticleID: <OTRS_TICKET_ArticleID>
  TicketNumber: <OTRS_TICKET_TicketNumber>
  # ...
```

It is possible to configure a URL for each item in the list, so the agent has the possibility to go to a website directly with just one click.

STORM comes a builtin invoker type called `Generic::ArticleMetaFilter` to be used in web services for this specific purpose. Only this type of invoker can be used for this functionality.

9.2 Usage

To properly display the results of the article meta filter requests it is highly recommended to add or extend the XSLT inbound mapping to include the list of results in tags called `Items` consisting in a single or multiple tags.

Example for one item:

```
<Items>Result 1</Items>
```

Example for more items:

```
<Items>Result 1</Items>
<Items>Result 2</Items>
<Items>Result 3</Items>
```

To search for IP addresses:

1. Create a web service with the XSLT mapping above to call an external server with the IP addresses. The web service should return a list of something, for example a list of host names associated to the passed IP addresses.
2. Create a new ticket.
3. Fill in the required fields.
4. Enter the following text in the body: *Your IP address is 192.168.0.1 and your subnet mask is 255.255.255.0*.
5. Go to the ticket detail view of the newly created ticket.

6. Expand the first article in the *Communication Stream* widget to see the buttons below the article.

The web service will search for all possible IP addresses in the article as configured by the regular expression and return a list of host names.

The buttons point to the search results of a web service. This should be returned the same search results when an agent calls the web service with the given IP addresses. The text for the buttons (*IP Address* in this example) comes from the `Label` key of the underlying system configuration setting.

If the agents hover the mouse over a button, they will get a preview of the list of results returned by the web service. Clicking on the buttons could open a URL associated to the results.

This feature works for all articles of a ticket.

CHAPTER

TEN

COLOR INDICATORS FOR DYNAMIC FIELD VALUES

If some field values are very important and must be immediately noticed, security analysts benefit from dynamic field dropdown and multiselect color definitions for each of the possible values. This allows users to focus on critical or urgent tasks with a glance.

With this function it is possible to add color indicators to the values of dynamic fields. This can help users to understand the impact or the criticality of the value.

To define color indicators for a dynamic field:

1. Go to the *Dynamic Fields* module in the administrator interface.
2. Add or edit a dynamic field of type *Dropdown* or *Multiselect*.
3. Define the values and assign a color to each value.

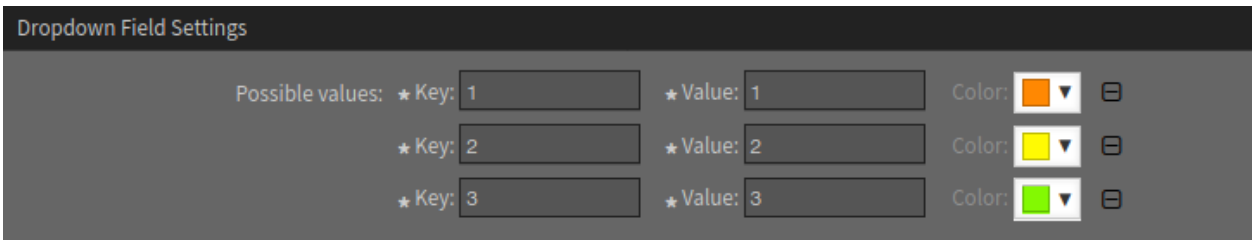

Fig. 1: Assigning Color Indicators

See also:

Please refer to the administrator manual how to display dynamic fields on screens.

The color indicators will be displayed for the configured dynamic field in each screen.

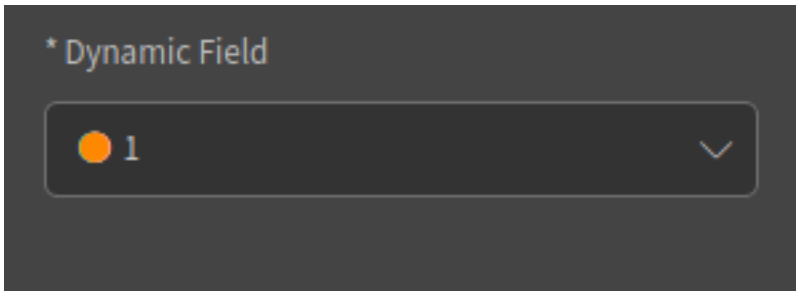

Fig. 2: Dynamic Field in Agent Interface

CHAPTER

ELEVEN

ENCRYPTION AUTO SELECT

With this function it is possible to reply to e-mails with an auto select of the signing and encryption method. The signing and encryption of the reply will be auto selected by using the same signing and encryption method as the incoming mail.

11.1 Requirements

The following requirements are needed to use the function:

- Configured PGP and/or S/MIME support.
- Added public and private PGP keys and/or certificates and private keys for S/MIME.
- Configured email address to fetch emails from.

See also:

Information of how to configure PGP and S/MIME can be found in the PGP Keys, S/MIME Certificates and Setting up Incoming Emails chapters of the administration manual.

11.2 Usage

The feature works for encrypted, signed or encrypted and signed articles.

To encrypt the reply of an article:

1. Open the detail view of a ticket and expand the encrypted article.
2. Click on the *Reply via Email* article action. Depending on the original message the field *Email Security* will be pre-filled with the corresponding method for signing and/or encryption.

The pre-selected options should not be reset if they are changed by the user after other fields are changed.

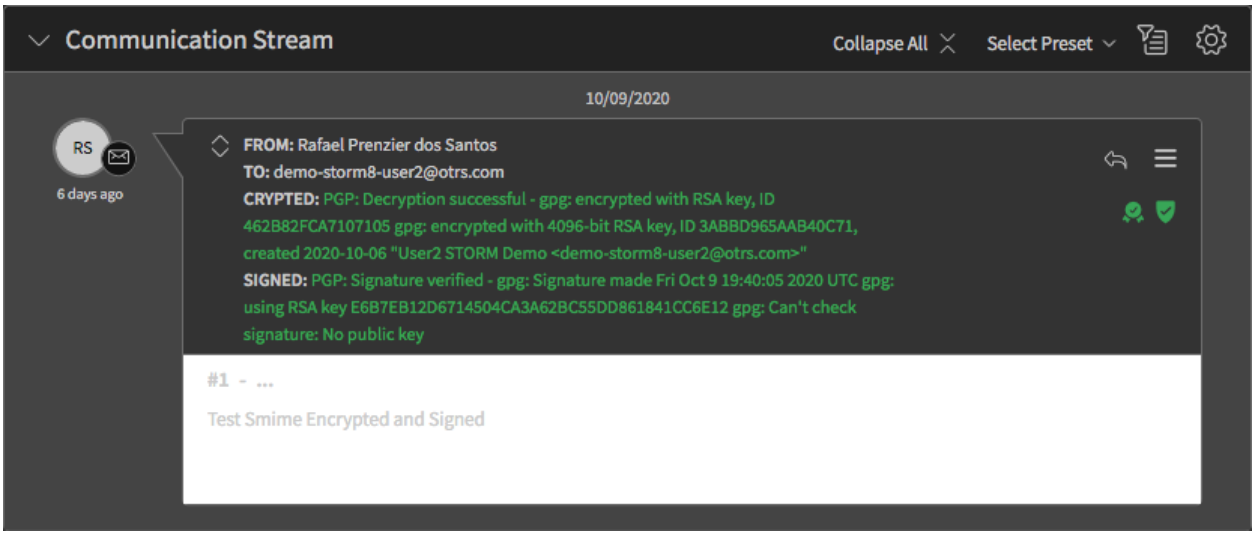

Fig. 1: PGP Signed and Encrypted E-mail

CHAPTER

TWELVE

DECRYPT BCC EMAILS

Security analysts benefit from decryption of incoming emails, even if the recipient address is in the blind carbon copy (*Bcc*) field because it allows them to decrypt mails that contain a STORM mail address as recipient in the blind carbon copy field.

12.1 Setup

To following setup is needed for using with **S/MIME**:

- The setting `SMIME::Decrypt::Methods###Email` searches for certificates that match email addresses inside the mail. This setting is enabled by default.
- The setting `SMIME::Decrypt::Methods###System` searches for certificates that match email addresses defined as system addresses. This setting is also enabled by default.
- The setting `SMIME::Decrypt::Methods###All` searches for all available S/MIME certificates to try to decrypt (brute force, disabled by default). It can be enabled for testing. In productive systems if the system has several certificates it is not recommended due to performance issues.

For **PGP** no additional settings are needed.

12.2 Usage

Send an email encrypted with PGP or S/MIME from your personal account to the email address configured in OTRS but using the blind carbon copy (*Bcc*) field only (do not fill in the *To* or the *Cc* field). Go to the ticket detail view of the new ticket and the article should be correctly decrypted.

Fig. 1: Decrypted Bcc Email Example

CHAPTER

THIRTEEN

EMAIL SECURITY

The `EnforceEmailSecurityRecipients` system configuration setting defines a list of email addresses to always force the encryption and/or signing. It is possible to use regular expression to match several addresses like `REGEX:(.*@example\.com)`.

The sender and all recipients for each email should be configured to use the same encryption engine either PGP or S/MIME. The system is not capable to mix them.

If the encryption of an email recipient is enforced, all recipients of this email must have a public key or certificate in the system. The email must be encrypted for all recipients, otherwise this could be considered a security issue.

If more than one key and certificate for the sender or a recipient exist in the system (if enforced), this function selects the first valid certificate. Except if another one has been previously specified in the user interface.

Note: The email sending will fail if the system could not find all the enforced keys and certificates.

If an agent uses PGP key or S/MIME certificate, the password reset email, the two-factor verification email, the ticket notification email and the appointment notification email can be sent signed and/or encrypted. PGP is favored over S/MIME.

To enable this feature, the relevant management screens have a *Send signed and/or encrypted email* checkbox. If this checkbox is selected, the email will be sent in signed and/or encrypted form.

CHAPTER

FOURTEEN

HARDWARE SECURITY MODULE (HSM) SUPPORT FOR PRIVATE KEYS

If controlled access to the server file system is not considered sufficient security, security managers benefit from hardware security module (HSM) support for private keys. Using this, they can authenticate against certificates and other cryptography operations that have an HSM installed or connected. This ensures more secure private keys and passwords, rather than storing certificates and their keys and password on the server's file system.

14.1 Requirements

- Configured email addresses for sending and receiving emails.
- Nitrokey HSM USB card for S/MIME:
  - Private key for the configured email stored in the card.
  - Certificate for the configured email stored in the card.
  - Configured S/MIME support in OTRS.
- Nitrokey Start USB card for PGP:
  - Private key for the configured email stored in the card.
  - Configured PGP support in OTRS.
  - Added public PGP key in OTRS.
- Installation of OpenSC tools:

```
opensc-tool
opensc-explorer
pkcs11-tool
pkcs15-tool
libp11
```

14.2 S/SMIME

This section describes how to use the NitroKey HSM card with S/MIME.

14.2.1 Preparation

In order to use the NitroKey HSM card with OTRS it is necessary to first configure OpenSSL to work with `libp11`. While there are different ways to do this it is recommended to create a custom configuration file and include at the beginning the main OpenSSL configuration file. An example of this custom configuration file is shown below:

```
openssl_conf = openssl_init

[openssl_init]
engines = engine_section

[engine_section]
pkcs11 = pkcs11_section

[pkcs11_section]
engine_id = pkcs11
dynamic_path = <libpkcs11_PATH>
MODULE_PATH = opensc-pkcs11.so
init = 0

[req]
distinguished_name = req_distinguished_name

[req_distinguished_name]
```

Where `<libpkcs11_PATH>` is the path to the engine module provided by `libp11` such as:

```
/usr/lib/ssl/engines/libpkcs11.so
/usr/local/lib/engines-1.1/libpkcs11.dylib
...
```

Depending on the version, the operating system and the way it was installed this can be different in your situation. Please refer to the documentation of `lib11` to find the correct paths according to your installation.

Save the file for example in `/etc/openssl/hsm.cnf` and include a link to this file in the beginning of the OpenSSL original configuration file as:

```
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#
# Note that you can include other files from the main configuration
# file using the .include directive.
# .include filename
.include /etc/openssl/hsm.cnf
```

14.2.2 Settings

1. Set the correct path and enable `SMIME::PKCS15ToolBin`. If the **pkcs15-tool** is correctly installed the path can be found by executing the `which pkcs15-tool` command. The result should be copied and pasted into the setting.
2. Set the correct path and enable `SMIME::HSMPrivatePath`. It is recommended to create a new directory different than `SMIME::PrivatePath` to prevent any confusions and left overs.
3. Set the correct HSM card serial number and user PIN in `SMIME::HSMCard::PIN`. The current card serial number can be obtained by using the `Maint::SMIME::HSMCard::Check` console command. The current user PIN is the one used in the initialization. The card serial numbers for example cards can be removed.
4. Enable the HSM card usage in setting `SMIME::UseHSM`.

14.2.3 Checking Environment

Execute the `Maint::SMIME::HSMCard::Check` console command. The output should look like this:

```
Reading HSM card information...
+-----------------+--------------------+
| Label | SmartCard-HSM                |
| Serial number   | DENK0100003        |
| Manufacturer ID | www.CardContact.de |
| User PIN        | Verified           |
+-----------------+--------------------+
Checking OpenSSL engines...
+--------------------------------+-------------+
| Dynamic engine loading support | Available   |
| pkcs11 engine                  | Available   |
| Intel RDRAND engine            | Available   |
+--------------------------------+-------------+
Done
```

It is important that the `User PIN` is verified and that the `pkcs11` engine is available. If any other listed engine is not available, it does not necessarily mean a problem.

14.2.4 Importing HSM Card Certificates and Keys

While signing and decryption of the messages is done in the HSM card it is still needed that OTRS imports certain information from the HSM card. The public certificates needs to be copied from the card to the file system. This can be done in the usual `SMIME::CertPath` setting.

A stub file with some private key meta data needs to be generated and placed in the path that was set in `SMIME::HSMPrivatePath` setting. This meta data includes the HSM card serial number, the key ID, label, hash, modulus, etc.

To accomplish this task, a new `Maint::SMIME::HSMCard::Sync` console command has been created. The output should look like this:

```
Syncronizing certificates and private keys metadata...
  Reading HSM card information... OK
  Reading HSM card objects... OK
  Processing HSM certificates and keys...
```

(continues on next page)

(continued from previous page)

```
  ID 'a1b2e3d4'... OK
  ID 'f5d6e7c8'... OK
Done.
```

14.2.5 Usage

The HSM card should now be ready for use and the usage should now be transparent for the users, e.g. creating a new email ticket.

Make sure to use a queue where the system address has a certificate and a private key in the HSM module and choose to sign the email with S/MIME from the security options field. The submitting of the form might be slightly slower as the HSM card needs to be unlocked and perform the operation.

14.3 PGP

This section describes how to use the NitroKey HSM card with PGP.

14.3.1 Settings

It is needed to set the card user PIN as the stored key password in the `PGP::Key::Password` setting. For example, if the key ID is *11223344* and the card user PIN is *123456*, create a new item in the setting and in the first part set *11223344* and then set *123456* as the value.

To get the key ID:

1. Open the *PGP Keys* module in the administrator interface.
2. Check the ID for the key in the *Key* column.

Note: If the system is already configured and working with this existing key, the setting should already contain an entry for the ID. In this case the key password should be already set, but it is still needed to exchanged for the card user PIN to work correctly.

14.3.2 Usage

The usage should now be transparent for the users, e.g. creating a new email ticket. Make sure to use a queue where the system address has a public and private key pair in the secure card and choose to sign the email with PGP from the security options field.

CHAPTER

FIFTEEN

LOGIN-LOGOUT LOG

In some situations it is necessary to have knowledge about the login and logout activities of the users. With this feature it is possible to see in the system log which users have been logged in and out.

This feature does not have any user interface, it only logs the activities in the system log. However, the *System Log* module of the *Administration* group in the administrator interface can be used to review the log entries.

15.1 Setup

The following system configuration settings have to be changed to enable the feature.

- `MinimumLogLevel` → *info*
- `UserLoginLogoutLog` → enabled

The following system configuration settings define an optional prefix for the log entries. This makes it easier to parse the log file.

- `UserLoginLogoutLog::LoginMessagePrefix`
- `UserLoginLogoutLog::LogoutMessagePrefix`

15.2 Usage

As an agent, login to the system and then logout. As an administrator, check the system log.

The login and logout data are displayed as log entries. If the prefixes for login and logout are defined then the entries contain this prefix.

```
Thu Oct 22 14:51:53 2020 (Europe/Berlin)   info   WebApp-10   LOGOUT_EVENT - Logout␣
↪by 'John Smith'.
Thu Oct 22 14:51:26 2020 (Europe/Berlin)   info   WebApp-10   LOGIN_EVENT - Login by
↪'John Smith'.
```

CHAPTER

SIXTEEN

DYNAMIC RECIPIENTS FOR TEMPLATES

Service agents can send emails and notes to several repeating recipients. They can save time in communications by predefined recipients in templates, rather than always adding repeating recipients from scratch, whenever messages must be send to a dedicated list of recipients.

This feature adds the possibility to use any ticket attribute in *To* and *Cc* field of a template via OTRS smart tags. With this feature, the administrator can add dynamic recipients for templates.

The recipient fields are already included in *Categories For Text Modules* feature. What STORM is added, to use OTRS smart tags in the recipient fields.

Note: This feature requires the *Categories For Text Modules* feature.

16.1 Usage

Example usage for dynamic recipients:

```
<OTRS_TICKET_State>@example.com
<OTRS_TICKET_DynamicField_ArchiveEmail>
support@<OTRS_TICKET_DynamicField_Company>.com
```

Depending of the ticket state, `<OTRS_TICKET_State>@example.com` will be replaced by *open@example.com*, *closed-successful@example.com* etc.

If a ticket dynamic field contains a full email address, the dynamic field can be used as an email address in the recipient fields.

To make the support request send to the appropriate company, `support@<OTRS_TICKET_DynamicField_Company>.com` can be used for this, if the dynamic field contains the name of that company.

Drop-down and multi-select dynamic fields are also supported.

Note: The type of ticket OTRS smart tags are supported only like `<OTRS_TICKET_...>`.

CHAPTER

SEVENTEEN

NOTIFICATION TEMPLATES

Traffic Light Protocol (TLP) designated email correspondence should indicate the TLP color of the information besides the TLP level in the body of the email, prior to the designated information itself.

In STORM this could be used for the notifications that are sent via email. For this purpose new templates have been added containing different layouts with colors indicating the status according to the traffic light protocol.

STORM comes with four pre-designed templates:

- `TLP-Red`
- `TLP-Amber`
- `TLP-Green`
- `TLP-White`

To set a TLP template for the ticket notification:

1. Go to the *Ticket Notifications* module in the administrator interface.
2. Select a notification from the list of notifications.
3. Select a TLP template for the email notification in the *Notification Methods* section.
4. Click on the *Save* or *Save and finish* button.

Depending on what is defined in the notification and what template has been assigned, the layout of the notification email will contain the chosen template.

[TICKET#2020101910000051] INCIDENT WAS IDENTIFIED

Dear Michael,

a new security incident was identified and classified. Please find the information below:

TLP Classification: TLP:RED
Classification: malicious-code::c2-server
Source of Event: SIEM
Event ID: 2020101910000013
Event Classification: Confirmed Attack with IR actions
Affected System: WS1254

Actions:
The System was taken down and will be analysed in a sandbox environnment

For further information please have a look at the incident in STORM

Fig. 1: TLP-designated Email Example

CHAPTER

EIGHTEEN

NOTIFICATION PLAIN TEXT EMAIL OPTIONS

This feature adds the possibility to send outgoing appointment notification and ticket notification emails in plain text. This can be useful whenever a remote system cannot read rich text emails.

18.1 Usage

To send the email notification as plain text:

1. Open the *Appointment Notifications* or the *Ticket Notifications* in the administrator interface.
2. Add a new notification or select an existing notification from the list of notifications.
3. Check the *Send email as plain text* field.

If the checkbox is checked, the rich text editor is changed to a regular text area in the *Notification Text* section. Only plain text can be entered in the text area for the content of the email.

Furthermore, the *Email template* field is selected to *Unformatted* and set to read-only. All emails that are sent by a such configured notification are in plain text format.

OTRS smart tags are still supported in the plain text format.

Notification Methods

These are the possible methods that can be used to send this notification to each of the recipients. Please select at least one method below.

Email

Enable this notification method: ☑

Additional recipient email addresses:
Use comma or semicolon to separate email addresses.
You can use OTRS-tags like <OTRS_TICKET_DynamicField_...> to insert values from the current ticket.

Article visible to customer: ☐
An article will be created if the notification is sent to the customer or an additional email address.

Create multiple articles: ☐
An article will be created for each additional recipient address of the notification.

Separator for recipients:
Use this setting to define a needed splitting symbol (e.g. ';' or ',').
This symbol is used as a separator for the addresses in the additional recipient addresses field.

Email template: Unformatted
Use this template to generate the complete email (only for HTML emails).

Enable email security: ☐

Email security level:

If signing key/certificate is missing: Skip notification delivery

If encryption key/certificate is missing: Skip notification delivery

Send email as plaintext: ☑
Email will be sent as plaintext

Web View

Enable this notification method: ☐

SMS (Short Message Service)

Please activate this transport in order to use it.

Fig. 1: Notification Methods Section

CHAPTER

NINETEEN

PDF IMAGE PREVIEW

This function allows to show an image preview of a PDF file in attachments for tickets and knowledge base articles. This is helpful to display the content of a PDF without downloading the file.

Additionally the package offers the possibility for an image preview in dynamic fields of type attachment (for PDF files).

Note: To use this feature, ImageMagick should be installed on the server that runs STORM.

19.1 Setup

1. Download and install ImageMagick. In the most recent versions of *ImageMagick*, the usage of PDF file is restricted. In order to allow the usage of PDF it is needed to update the *ImageMagick* configuration manually.
2. Search for the file `/usr/local/etc/ImageMagick-7/policy.xml` (file path might vary depending on the *ImageMagick* version).
3. Check if the file contains a `PDF` entry and that this entry is not marked as comment.
4. Check if the entry contains at least `read` rights.

```
<policy domain="coder" rights="read" pattern="PDF" />
```

5. Go to the system configuration and search for the setting `Magick::Bin`.
6. Acivate the setting and enter the file path to the *ImageMagick* binary.
7. Open the `$OTRS_HOME/Kernel/Config.pm` and add the programs to the allow list.

```
$Self->{'SystemConfiguration::ValueType::SystemCommand::BinaryWhiteList'}->{'001-
↪OTRSSTORM'} = [
    'magick',
    'pkcs15-tool',
];
```

8. Rebuild the system configuration.

19.2 Usage

After installation of the package, the *Attachments* widget now displays an icon for the PDF files in the *Preview* column in addition to the regular ones for images, audio and video files.

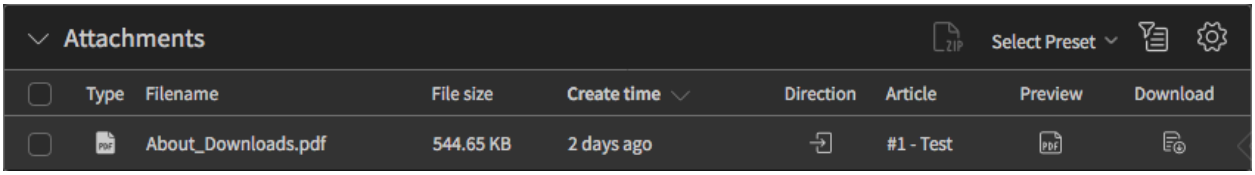

Fig. 1: Preview Column in Attachments Widget

To see an image preview of the PDF:

1. Open the ticket detail view or the knowledge base article detail view.
2. Click on the preview icon for a PDF file in the *Attachments* widget.

The PDF is now displayed as an image in a small preview window. In case of, that the file is not really a PDF, the preview window displays nothing. In this case it is recommended not to download the PDF.

Note: The displayed image of the PDF contains only the first page of the PDF.

19.2.1 Dynamic Fields of Type Attachment

If the *Dynamic Field Attachment* feature is installed and a dynamic field has been added to a property card, the dynamic field contains an additional preview icon next to the download icon.

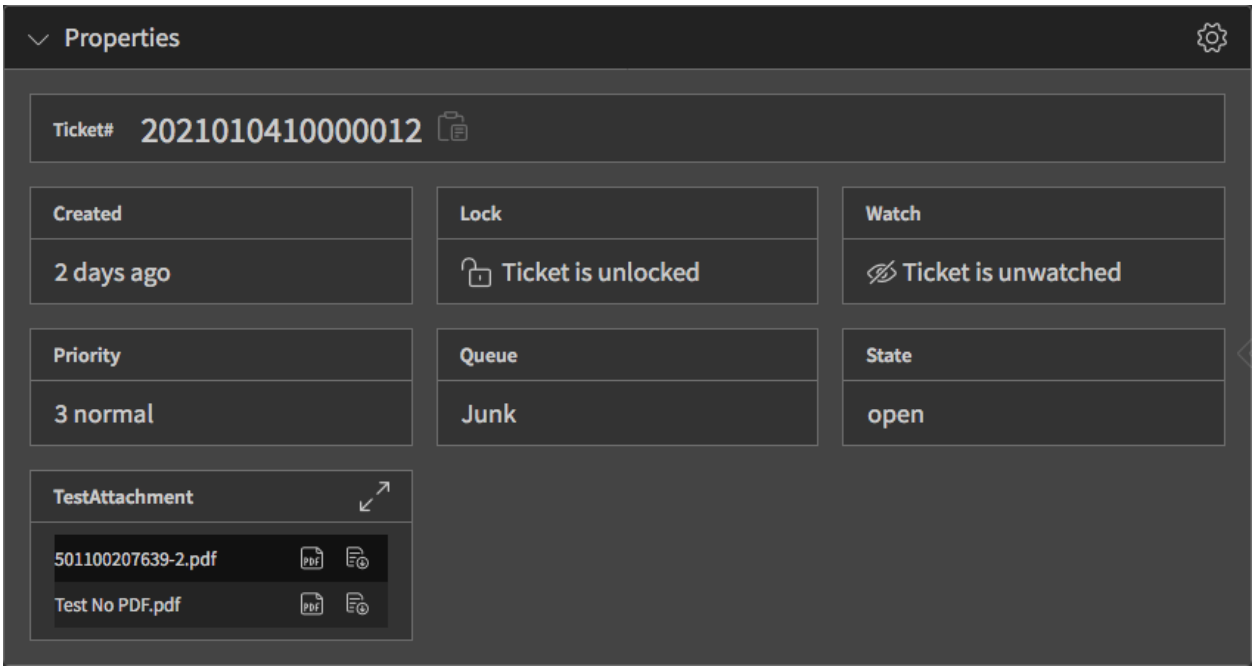

Fig. 2: Property Card with Dynamic Field of Type Attachment

The PDF image preview can be used with this feature whereever a dynamic field of type attachment is added to the system.

CHAPTER

TWENTY

PROCESS MANAGEMENT DIRECT ACTIONS

Any process has activity dialog and there are some fields in this activity dialog. The idea of the direct actions is to avoid unnecessary actions. If the process has a field with pre-defined value, and the agent does not need to do anything but just click on a button to submit the form, this action can be done automatically.

Direct actions work with all processes that use an activity dialog set as a direct action. However, there are some requirements:

- All fields in the activity dialog needs to be hidden.
- All fields in the activity dialog needs to have a default value.

There are some fields like *Queue*, *Priority* or *State* that already have pre-defined values in the configuration of the process management. If the administrators would like to specify another value, then they need to have a default value.

20.1 Example Usage

In this example we will define a very simple process with one activity and two activity dialogs. The first activity dialog allows to set the title of the ticket to any text, the second activity dialog sets a pre-defined text to the title of the ticket. This is called *direct action*.

The user task activity dialog is extended with a new field *Direct Action*. If this field is checked, the activity dialog will be submitted automatically.

Direct actions require that all fields should be manually set to hidden and provide a default value.

Do not forget to deploy the process once it is ready.

Now go to the agent interface and create a process ticket. Select the newly created process. The ticket detail view will show the two buttons that we defined in our very simple process.

The first button opens an action to set the title of the ticket to any text. This works the same as the feature has in the OTRS framework. The agent has to change the title of the ticket manually and then the form has to be submitted with the *Submit* button.

The second button has a flash icon, which means this is a *direct action*. If the agent clicks on this button, the title of the ticket is set to the text defined in the process and the action will be submitted automatically. No other action is needed manually by the agent.

The process can contain some triggers to go to one activity to another by setting any ticket field like state, queue or any dynamic field, by using the predefined direct actions. The users do not need to set any values to jump to another activity. With this feature, it is possible to add *Previous* or *Next* buttons to the dialogs of the process to jump forward or backward from one user activity.

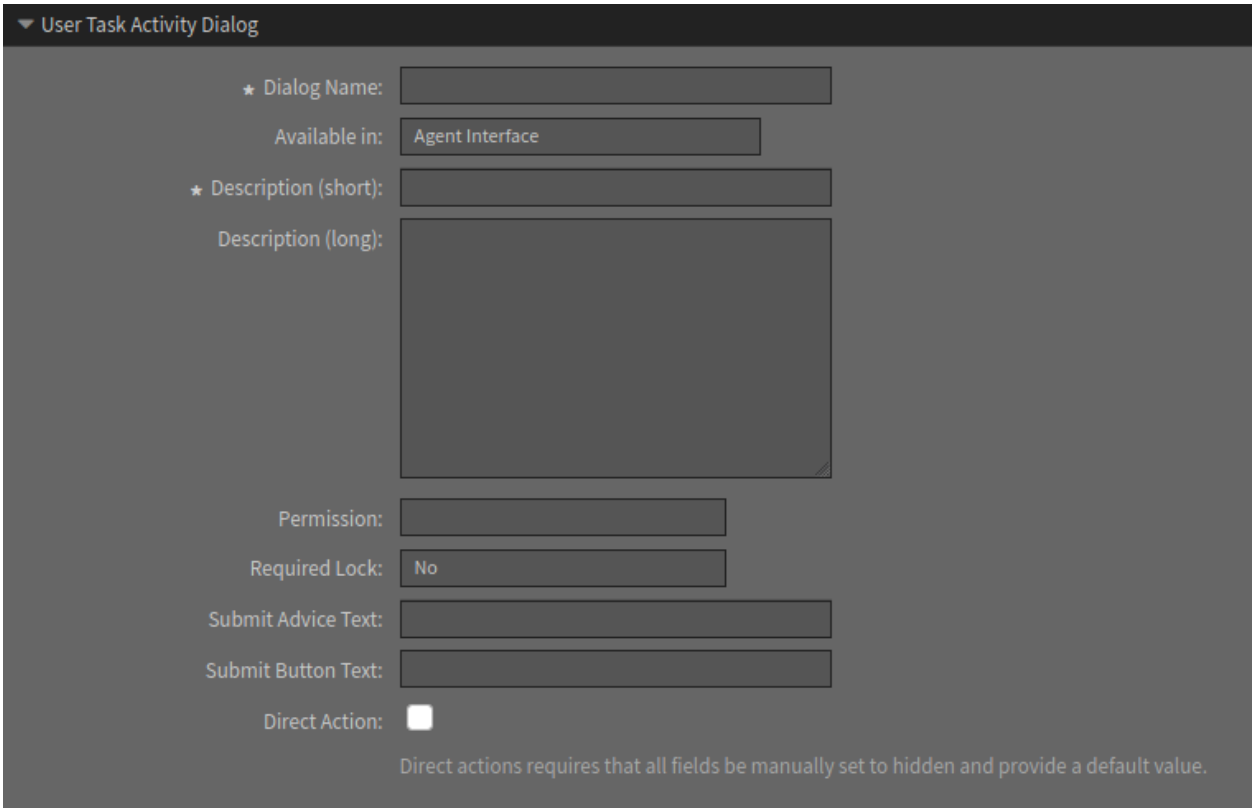

Fig. 1: Edit User Task Activity Dialog Window

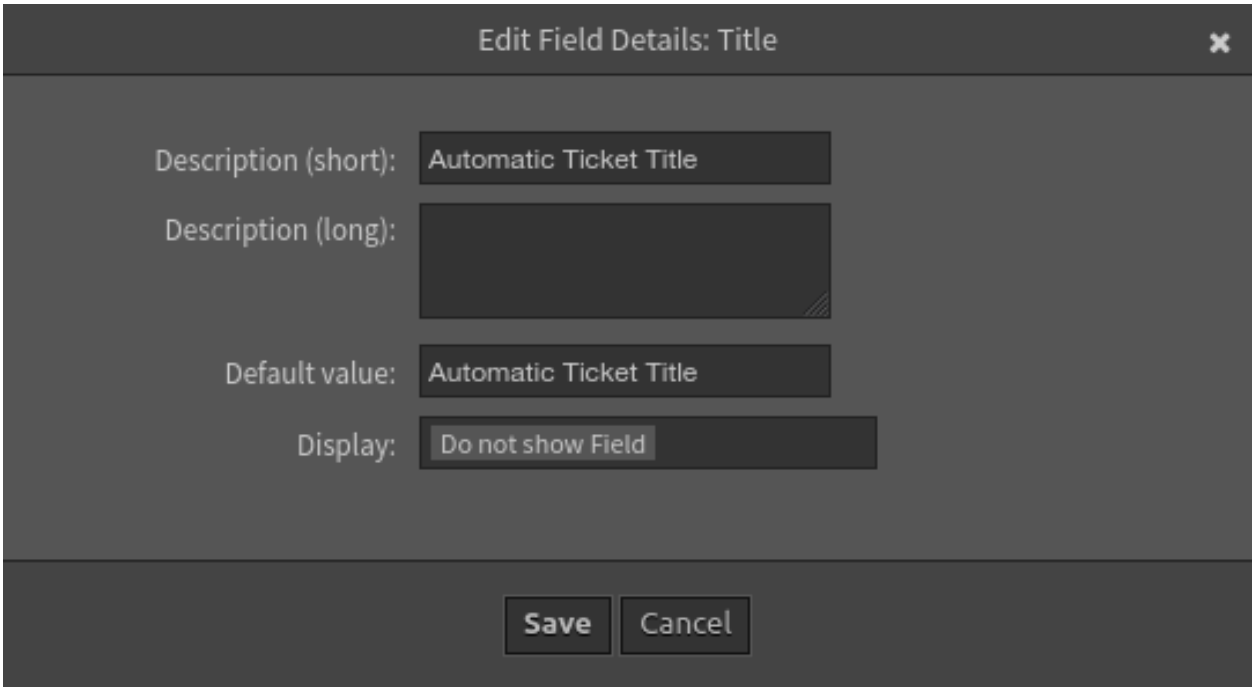

Fig. 2: Edit Field Details Dialog

Fig. 3: Edit Field Details Dialog

CHAPTER

TWENTYONE

PROCESS TASK ACTIVITIES ENCRYPTION AND SIGNING

With this feature it is possible to configure outgoing emails from process task activities to be signed and/or encrypted via S/MIME or PGP. This can be helpful if the recipients of a business process need encrypted communication.

21.1 Usage for Script Task Activities

To setup a script task activity with signing and/or encryption:

1. Go to the *Process Management* screen in the administrator interface.
2. Create or edit an existing process.
3. Create or select a script task activity and go to the configuration window.
4. Select the *TicketSendEmail* module and save.
5. Click on the *Configure* button next to the selected module.
6. Select one of the signing and encryption methods in the *Email Security* section.

All sent emails from such configured script task activity will now follow the selected security options with the corresponding exceptions.

21.2 Usage for User Task Activities

Note: This feature requires the *Process Management Article Email* feature.

To setup a user task activity with signing and/or encryption:

1. Go to the *Process Management* screen in the administrator interface.
2. Select a user task activity dialog and go to the configuration window.
3. Select the *Article* field in the *Available Fields* section and move it to the *Assigned Fields* section.
4. Select *Email* in the *Communication Channel* field.
5. Save and deploy the process.
6. Go to the agent interface and create a new process ticket.

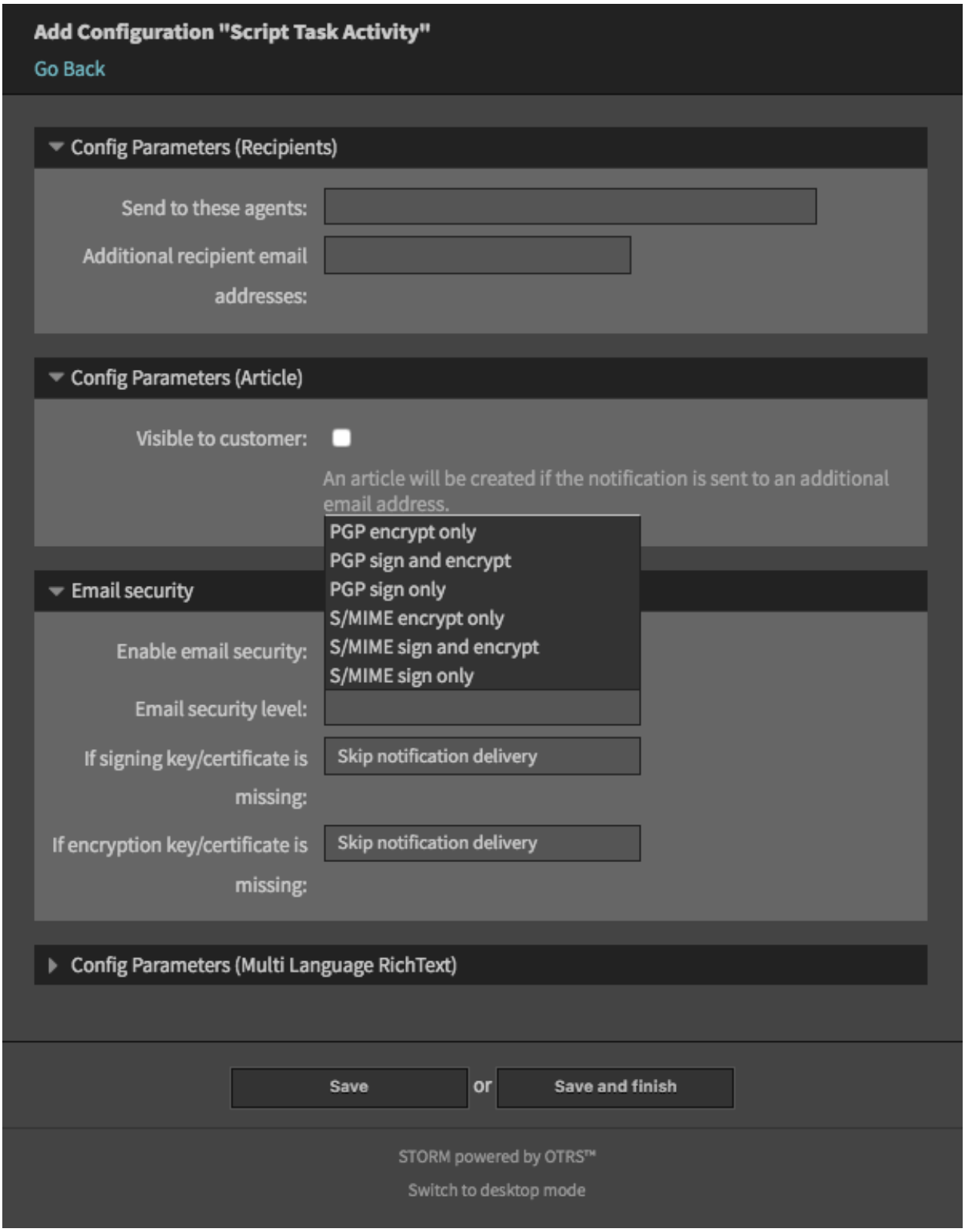

Fig. 1: Script Task Activity Configuration Window

7. A new *Email Security* field is added to the *Articles* section of the process ticket to select the signing and/or encrypting possibilities of the email.

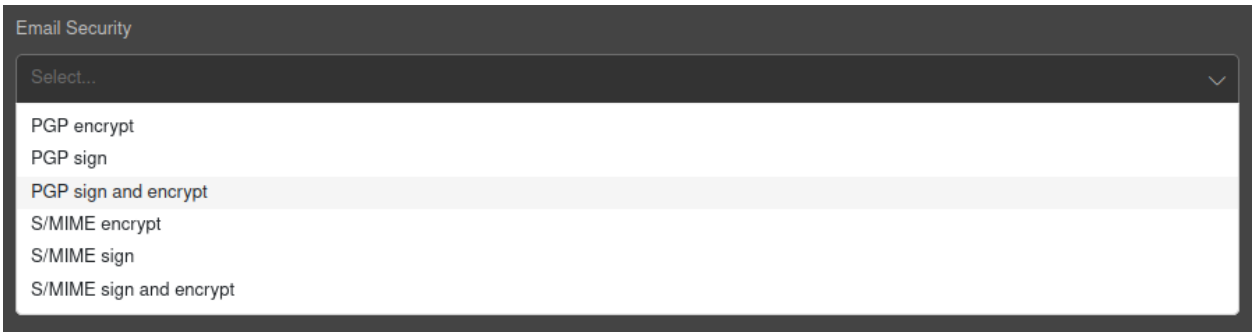

Fig. 2: Email Security Field in Process Ticket

CHAPTER

TWENTYTWO

PROCESS MANAGEMENT MODULE SYSTEM CALL

When a process moves from one activity to another activity, an action can be attached to the sequence flow. These actions are defined as modules to perform specific task like change ticket attributes, create articles, set dynamic fields, etc. The modules can also be attached to certain process management activities which are called *script task activities* and execute the attachment module when they are reached.

The system call module allows the agents to call any program, script, binary or executable that is available in the operating system of the server running OTRS. The result of the system call can be used to update the ticket information.

The system call module requires the use of XSLT mappings. Outbound mapping is used to define the system command to be called, and inbound mapping is used to convert the results from the system call to update the current ticket.

In the outbound mapping, it is necessary to have the `<Command>` key and if needed one or more `<Argument>` keys. The values to set can be transformed from the process ticket under the `<Ticket>` key and then the normal ticket attributes as sub-keys such as `<Priority>`, `<QueueID>`, `<Title>` etc. Or it could be defined as fixed values.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <xsl:copy>
            <RootElement>
                <Command>command</Command>
                <Arguments>argument1</Arguments>
                <Arguments>argument2</Arguments>
                <Arguments>argumentN</Arguments>
                <Arguments><xsl:value-of select="//Ticket/Priority"/></Arguments>
            </RootElement>
        </xsl:copy>
    </xsl:template>
</xsl:stylesheet>
```

For security reasons, only those commands can be added to the `<Command>` key, that are added to the `ProcessManagement::Modules::SystemCall::CommandWhiteList` setting as allowed commands. This will prevent users to run not allowed commands on the server.

The inbound mapping is used to convert the results from the system call in information to update the current ticket. All subkeys must be inside the `<Ticket>` key. Here is the possible list of subkeys:

```
<CustomerUser>
<DynamicField>
<Lock>
<LockID>
<Owner>
```

(continues on next page)

(continued from previous page)

```
<OwnerID>
<Pending>
<Priority>
<PriorityID>
<Queue>
<QueueID>
<Responsible>
<ResponsibleID>
<Service>
<ServiceID>
<SLA>
<SLAID>
<State>
<StateID>
<Title>
<Type>
<TypeID>
```

The result values can be accessed from:

<ReturnCode>
The numeric value returned from a system process execution.

<Output>
Any text printed to the standard output.

<ErrorOutput>
Any text printed to the standard error output.

Here is an example inbound mapping, which sets the output of the system call as ticket title:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <xsl:copy>
            <RootElement>
                <Ticket>
                    <Title><xsl:value-of select="//Output" /></Title>
                </Ticket>
            </RootElement>
        </xsl:copy>
    </xsl:template>
</xsl:stylesheet>
```

See also:

There is a *Mapping handling explanation* section in the configuration screen. This explanation can be used as a reference.

The system calls are executed by the OTRS daemon in the background with the proper permissions asynchronously. If a system call takes some time, the process management will wait until the system call is terminated and the results of the system call are ready. During this period the process cannot be proceeded to the next state, but the other part of the agent interface still can be used.

22.1 Example Usage

In this example we will define a very simple process with one script task activity. The process is configured to change the title of the ticket to the result of the system command `uname -s`. The result could be *Darwin*, *Linux*, *GNU* etc, depending on the operating system.

To define an example process:

1. Go to the process management screen and create a new process.
2. Add a new script task activity to the process.
3. Select `SystemCall` in the *Script* field of the *Script Settings* section. Click on the *Save* button.
4. Click on the *Configure* button next to the *Script* field.
5. Add the following lines to the *Outbound: XSLT Mapping* template.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <xsl:copy>
            <RootElement>
                <Command>uname</Command>
                <Arguments>-s</Arguments>
            </RootElement>
        </xsl:copy>
    </xsl:template>
</xsl:stylesheet>
```

6. Add the following lines to the *Inbound: XSLT Mapping* template.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <xsl:copy>
            <RootElement>
                <Ticket>
                    <Title><xsl:value-of select="//Output" /></Title>
                </Ticket>
            </RootElement>
        </xsl:copy>
    </xsl:template>
</xsl:stylesheet>
```

7. Click on the *Save and finish* button.
8. Deploy the process.
9. Create a new process ticket in the agent interface, then click on the activity *Start* button.
10. Wait until the daemon executes the system call.
11. The ticket title is changed to the result of `uname -s`.

CHAPTER

TWENTYTHREE

SHARED TICKET WATCHLISTS

This feature allows to tag tickets with labels, so that service agents can save a lot of time when searching for already analyzed incidents or service requests, rather than wasting time for analyzing requests but not tagging them according to the analysis results, whenever tickets must be classified.

Note: This feature requires the *Ticket Watchlist* feature.

23.1 Usage

This is an enhancement for the *Ticket Watchlist* feature and allows to share the watchlists not only to the deputies, but to anyone in the system.

See also:

See the Ticket Watchlist documentation in the *Features Manual* for the base functionality.

The ticket watchlist overview can be accessed via the glasses icon in the organizer sidebar.

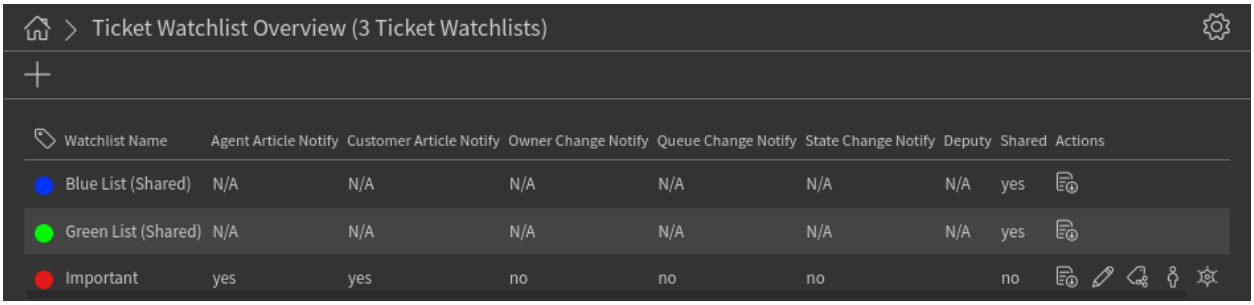

Fig. 1: Ticket Watchlist Overview

The example above shows two shared watchlists and a personal watchlist. The third icon in the *Actions* column makes possible to share a personal watchlist.

The shared watchlists have no notifications for other agents and they cannot see the deputies of the shared watchlists. Other agents can only export the ticket list and add or remove tickets using the ticket detail view.

If the agent is promoted as a deputy of the watchlist, a police badge icon will indicate this. The deputy has some more privileges for the watchlist.

See also:

The deputy feature is described in the *Ticket Watchlist* documentation.

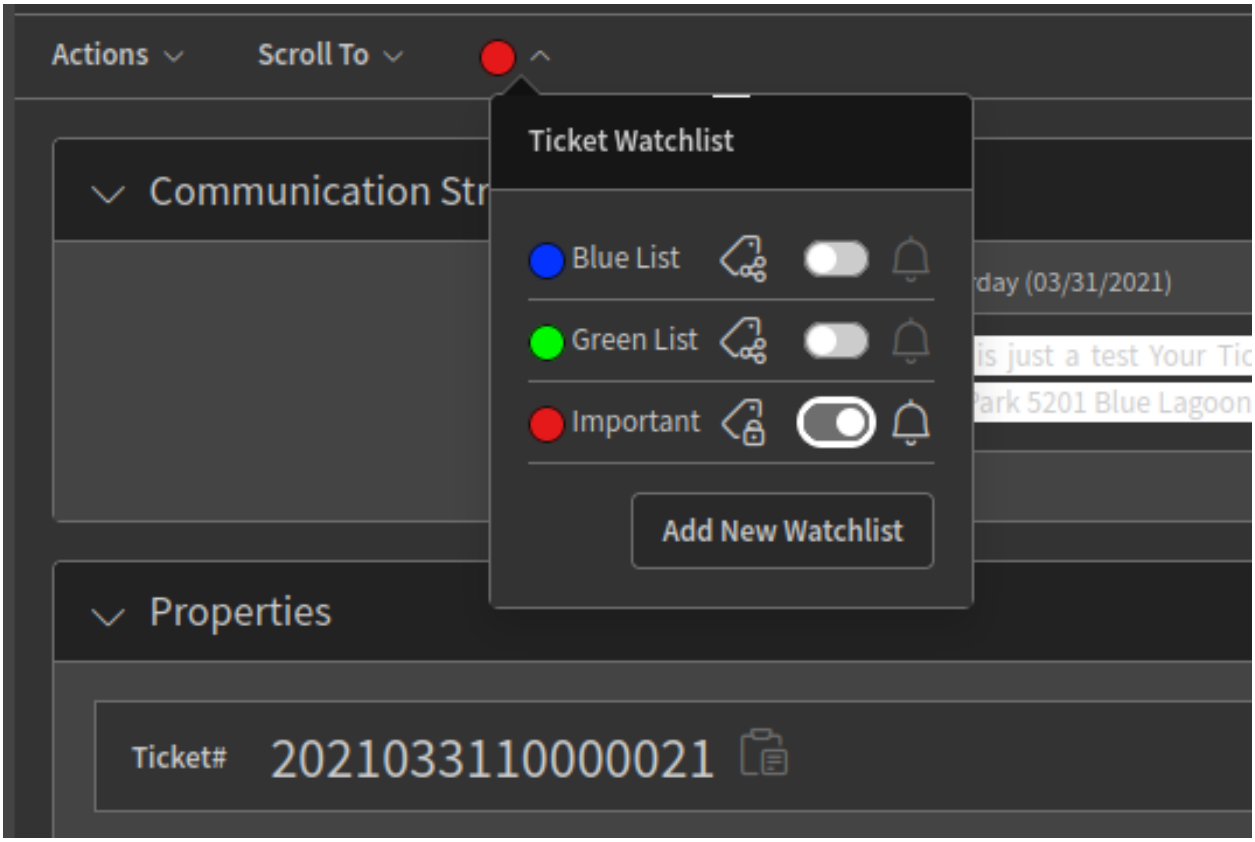

Fig. 2: Ticket Watchlist in Ticket Detail View

The lock icon in the bottom right corner of the tag icon indicates that the watchlist is personal. If there is an icon consisted of three circles it means that the watchlist is shared.

Other agents have basically read-only permission for shared watchlists, but they could add or remove tickets to the shared watchlists depending on the ticket permissions (*rw* by default). This means that they have read-write permission for the specific ticket. Otherwise if an agent has only read-only permission for a ticket, the agent cannot remove it from the shared watchlist.

Permission level to grant add or remove tickets to a shared watchlist can be defined in the `AgentFrontend::SharedTags::Ticket::PermissionType` setting.

CHAPTER

TWENTYFOUR

TAGGING LABELS FOR ATTACHMENTS

This feature allows to tag attachments with labels, so that security analysts can save a lot of time when searching for already analyzed attachments, rather than wasting time for analyzing attachments but not tagging them according to the analysis results, whenever attachments must be classified.

24.1 Usage

This feature has similar functionality as *Shared Ticket Watchlists* but for attachments.

The attachment tag overview can be accessed via the attachment icon in the organizer sidebar.

Fig. 1: Attachment Tag Overview

It is possible to edit, share and delete an attachment tag as well as change the owner. If the attachment tag is shared, it will be visible to any agents in the system.

When an agent creates an attachment tag, the tag can be added to the attachments in any *Attachments* widget in a business object detail view.

Permission level to grant add or remove attachments to a shared tag can be defined in `AgentFrontend::SharedTags::Attachment::PermissionType` setting.

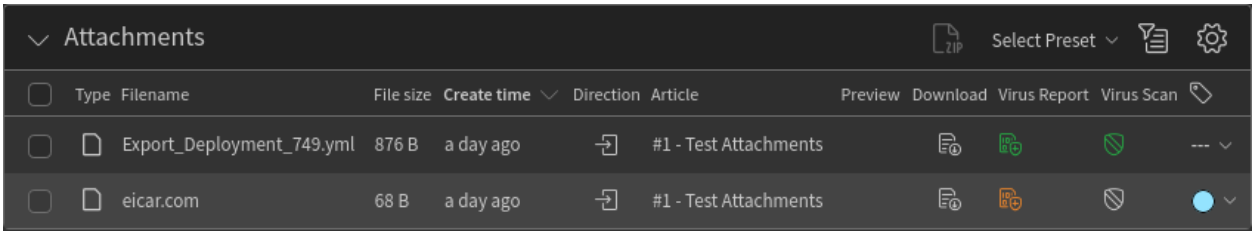

Fig. 2: Attachments Widget

24.2 Using Attachment Tags as Ticket Filter

The attachment tags can be also used as a ticket filter. They work as any other filter in a ticket list. If you use the filter, you can select one or more attachment tag names (that you have access to).

If a ticket has at least one article attachment that belongs to one of the selected attachment tags, this ticket will be displayed in the filtered list.

CHAPTER

TWENTYFIVE

TAXONOMY

25.1 Background

STORM provides predefined fields for Enisa, KRITIS and TLP taxonomies and event classification.

- Enisa - *European Union Agency for Cybersecurity*
- KRITIS - *Kritische Infrastrukturen*
- TLP - *Traffic Light Protocol*

New dynamic fields are added to enhance tickets with standardized information and data:

```
- EnisaSecurityIncidentClassification
- EventClassification
- KRITISSituationAssessment
- KRITISTaxonomy
- TLP
```

Some of these dynamic fields are drop-down fields, while others are dynamic fields of type web service. The dynamic fields are marked as internal, so they cannot be deleted from the system.

In order to update their values easily, some of these fields use web services internally, that by default does a loop-back request to a specific static file that is installed in the local system. The web services can be modified to point to another server that could provide an updated file that could be maintained by third parties.

STORM provides not only the dynamic fields, but also the underlying web services and the needed static files that are the source of the information for some of this dynamic fields. The web services point to the local server `localhost:8080` and the related dynamic fields have a default caching configuration of their values set for 1 day (86400 seconds).

Please make sure that the configuration of the web services is synchronized with the OTRS web server. Any change will require to cleanup the cache by executing the OTRS console command `Maint::Cache::Delete`.

The static files are saved in `<OTRSHOME>/httpd/htdocs/STORM`. A direct file modification is not possible. In order to modify you need to copy the file and save it with another name. The new file can then be customized. Please be aware to update the affected web service to point to the correct file.

Note: This feature requires the *Dynamic Field Web Service* feature.

If the default settings are not applicable for the current system, they can be changed in the system configuration and/or in the web service management screens.

25.2 Usage

All dynamic fields are added to the *Change Free Fields* action of the ticket detail view.

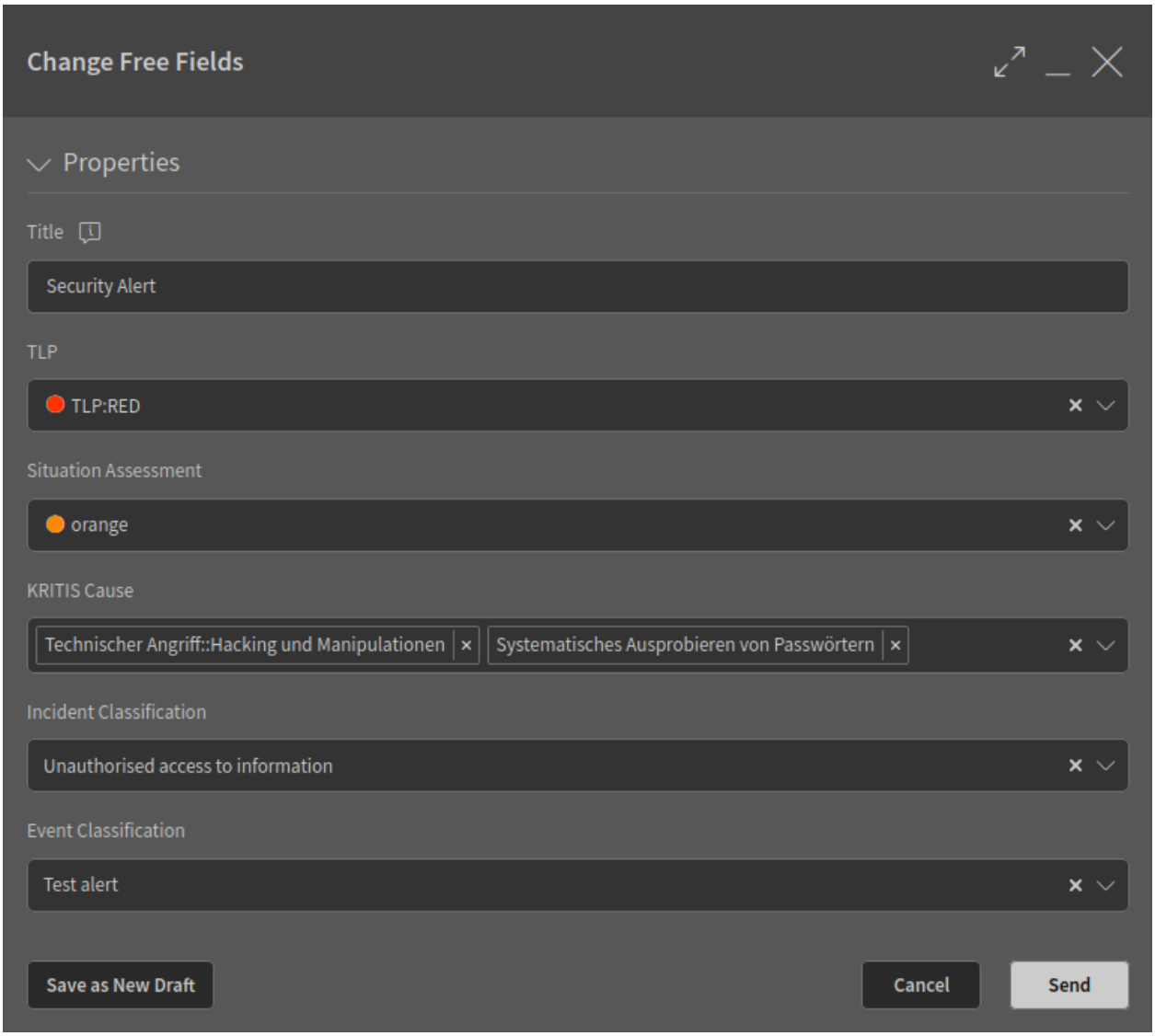

Fig. 1: Dynamic Fields in Change Free Fields Action

Furthermore, the TLP field is added to the *Properties* widget of the ticket detail view, and shown as a column in ticket lists and organizer items. The TLP field is set for inline editing when it is shown as column.

The settings can be modified to show more or less fields depending on the needs for a particular system.

See also:

Taxonomies are available in OTRS statistics.

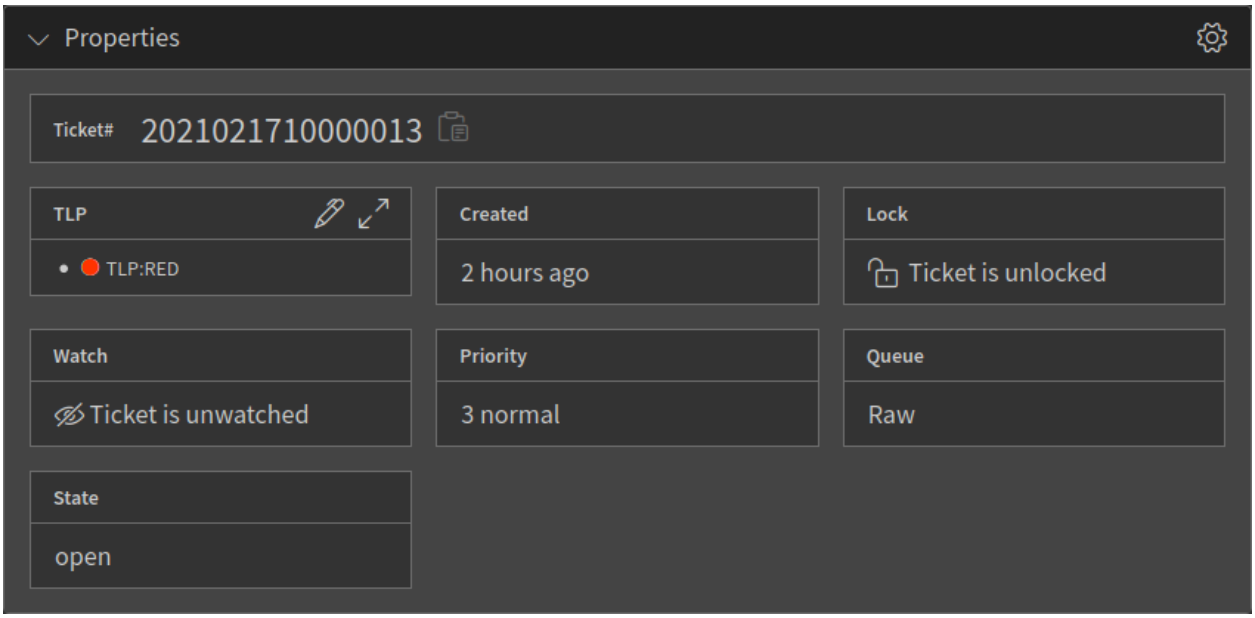

Fig. 2: TLP Dynamic Field in Ticket Properties

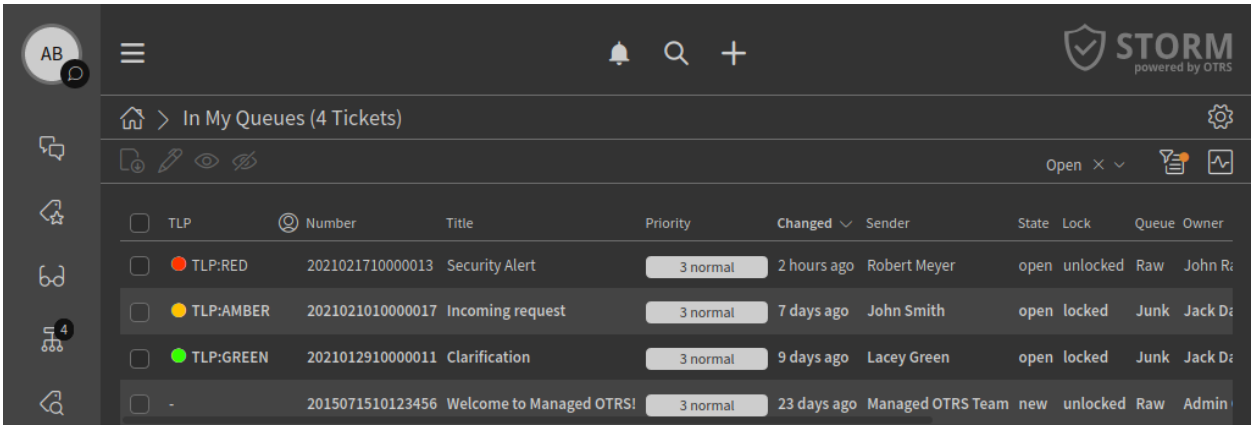

Fig. 3: TLP Dynamic Field in Ticket List

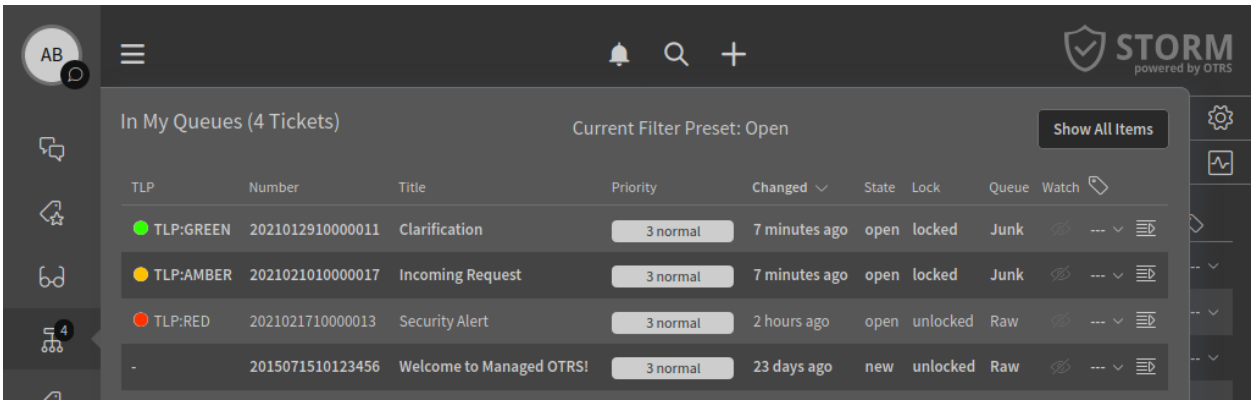

Fig. 4: TLP Dynamic Field in Organizer Item

CHAPTER

TWENTYSIX

PROCESSES

STORM comes with built-in processes for event triage, incident handling and task handling. All processes are invalid by default, and they have to be activated by an administrator first. However, each process has dependencies like ACLs, dynamic fields, queses, ticket notifications, ticket types, which have to be activated before the process itself has been activated.

This chapter explains how to make the processes work.

26.1 Setup

The processes can be activated in the *Process Management* screen of the administrator interface. All processes are *inactive* by default.

To activate the *Event Triage* process:

1. Go to the *Queues* screen of the administrator interface.
2. Set the `Incidents` queue to *valid*.
3. Go to the *Dynamic Fileds* screen of the administrator interface.
4. Set the following dynamic fields to *valid*.
   - `EventClassification`
   - `IncidentTicket`
   - `ProcessHelper`
5. Go to the *Access Control Lists (ACL)* screen of the administrator interface.
6. Set the following ACLs to *valid*.
   - `Event 001 - Forbid Actions`
   - `Event 001 - Forbid ActionsLimit DF Event Classification`
7. Deploy all ACLs.
8. Go to the *Process Management* screen of the administrator interface.
9. Set the `Event Triage` process to *valid*.
10. Deploy all processes.

To activate the *Incident Handling* process:

1. Go to the *Types* screen of the administrator interface.
2. Set the following types to *valid*.

- `Event`
- `Incident`
- `Task`

3. Go to the *Dynamic Fileds* screen of the administrator interface.
4. Set the following dynamic fields to *valid*.
   - `AnalysisResult`
   - `EnisaSecurityIncidentClassification`
   - `ISO`
   - `KRITISSituationAssessment`
   - `KRITISTaxonomy`
   - `LessonsLearned`
   - `ProcessHelper`
   - `RemediationAdvice`
   - `SendAdvice`
   - `TaskBody`
   - `TaskName`
   - `TaskRecipient`
   - `TaskResult`
   - `TaskSubject`
   - `TechContact`
   - `TLP`
5. Go to the *Ticket Notifications* screen of the administrator interface.
6. Set the following ticket notifications to *valid*.
   - `Incident: Send Mitigation & Remediation Advice - TLP Amber`
   - `Incident: Send Mitigation & Remediation Advice - TLP Green`
   - `Incident: Send Mitigation & Remediation Advice - TLP Red`
   - `Incident: Send Mitigation & Remediation Advice - TLP White`
7. Go to the *Access Control Lists (ACL)* screen of the administrator interface.
8. Set the following ACLs to *valid*.
   - `Incident 001 - Hide Actions and Dialogues`
   - `Incident 002a - Show Next Button in Analysis phase step 1`
   - `Incident 002b - Show Next Button in Analysis phase step 2`
   - `Incident 003a - Show Next Button in Mitigation phase step 1`
   - `Incident 003b - Show Next Button in Mitigation phase step 2`
   - `Incident 004 - Show close button`
   - `Incident 005 - Hide Kritis Taxonomy`

    - `Incident 005 - Show Kritis Taxonomy`
9. Deploy all ACLs.
10. Go to the *Process Management* screen of the administrator interface.
11. Set the `Incident Handling` process to *valid*.
12. Deploy all processes.

To activate the *Task Handling* process:

1. Go to the *Types* screen of the administrator interface.
2. Set the following types to *valid*.
    - `Incident`
3. Go to the *Dynamic Fileds* screen of the administrator interface.
4. Set the following dynamic fields to *valid*.
    - `TaskName`
    - `TaskResult`
5. Go to the *Access Control Lists (ACL)* screen of the administrator interface.
6. Set the following ACLs to *valid*.
    - `Task 001 - Hide Actions`
7. Deploy all ACLs.
8. Go to the *Process Management* screen of the administrator interface.
9. Set the `Task Handling` process to *valid*.
10. Deploy all processes.

26.1.1 Console Command

There is a console command to list, enable and disable the process groups. Execute the command with the `--help` option for detailed instructions about how it works.

```
$ bin/otrs.Console.pl Maint::STORM::ProcessGroups::Toggle --help

Enable/Disable a process group and its dependencies

Usage:
 otrs.Console.pl Maint::STORM::ProcessGroups::Toggle [--name ...] [--list] [--enable]␣
↪[--disable]

Options:
 [--name ...]                   - Name of the process group (all if omitted).
 [--list]                       - List all process groups.
 [--enable]                     - Enable the process group.
 [--disable]                    - Disable the process group.
 [--help]                       - Display help for this command.
 [--no-ansi]                    - Do not perform ANSI terminal output coloring.
 [--quiet]                      - Suppress informative output, only retain error␣
↪messages.
```

26.2 Usage

We developed the processes based on best practices. We also know that every customer has different workflow, so it might be that the processes have to be customized before deploy them and use them in production. Please consult with our experts before activating a process.

The general usage of processes are explained in the *Administrator Manual*. For detailed usage of the process above ask the *Customer Solutions Team*.

CHAPTER

TWENTYSEVEN

OFFLINE REGISTRATION

This feature is provided by an additional package. It allows to register the system in an offline environment.

Note: This package is not part of the default STORM installation. Contact OTRS Group to get a special contract to use this feature.

The following system configuration settings need to be enabled and set properly:

- `SMIME`
- `SMIME::Bin`
- `SMIME::CertPath`
- `SMIME::PrivatePath`

For the offline registration, a registration key file provided by OTRS Group is required.

27.1 Usage

The offline registration process can be done by either using the administrator interface via the browser or executing a command in the terminal.

To register the system via the administrator interface:

1. Go to the *System Registration* screen of the administrator interface.
2. Upload the registration key file provided by OTRS Group.
3. Click on the *Register* button to finish the registration process.

To register the system via the command line:

1. Save the registration key file provided by the OTRS Group somewhere in the system where the `otrs` user have read access.
2. As the `otrs` user, open the terminal and navigate to the home folder of STORM.
3. Execute the following command:

```
otrs> bin/otrs.Console.pl Admin::OfflineRegistration::LoadKey --key-path PATH_TO_
↪KEY_FILE
```

A contract is attached to the registration key file. This contract might contain an expiration date. Some days before it expires the system will show warning messages about it (same mechanism as with an online

registration). To continue use the system without any interruption it is necessary to contact OTRS Group to renew the contract and get a new registration key.

Any new registration key can be loaded into the system by using any of the methods described above at any time within the current contract expiration date.

Note: After a contract is expired the graphical user interface of STORM will be restricted and it can not be used. In such case is needed to use the command line method to load the new registration key file and renew the contract.