STORM Manual

Release 8.0

OTRS AG

Jan 20, 2021

Contents

CHAPTER 1

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.

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

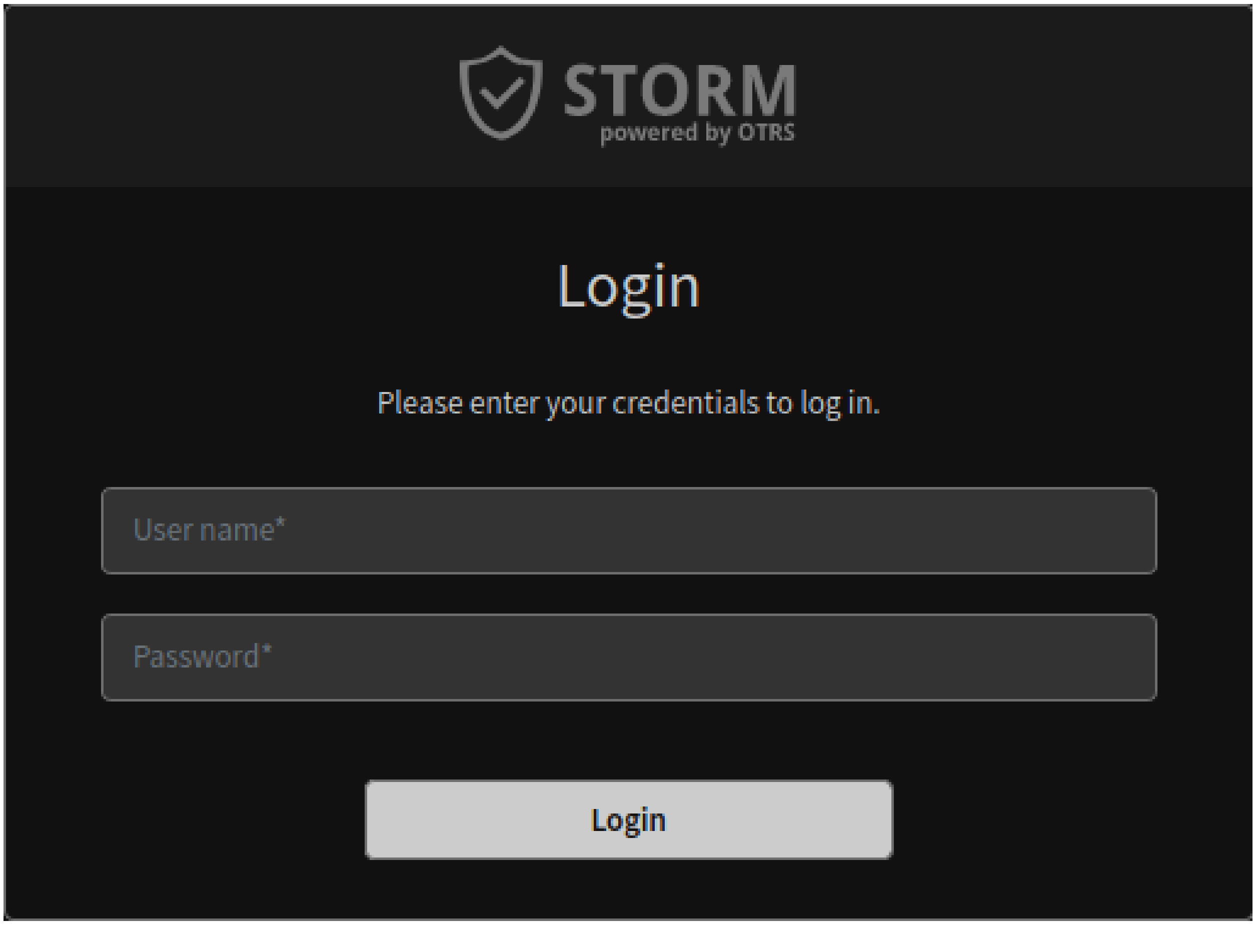
See also:

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

The administrators can change the skin in the agent preferences.

To change the skin:

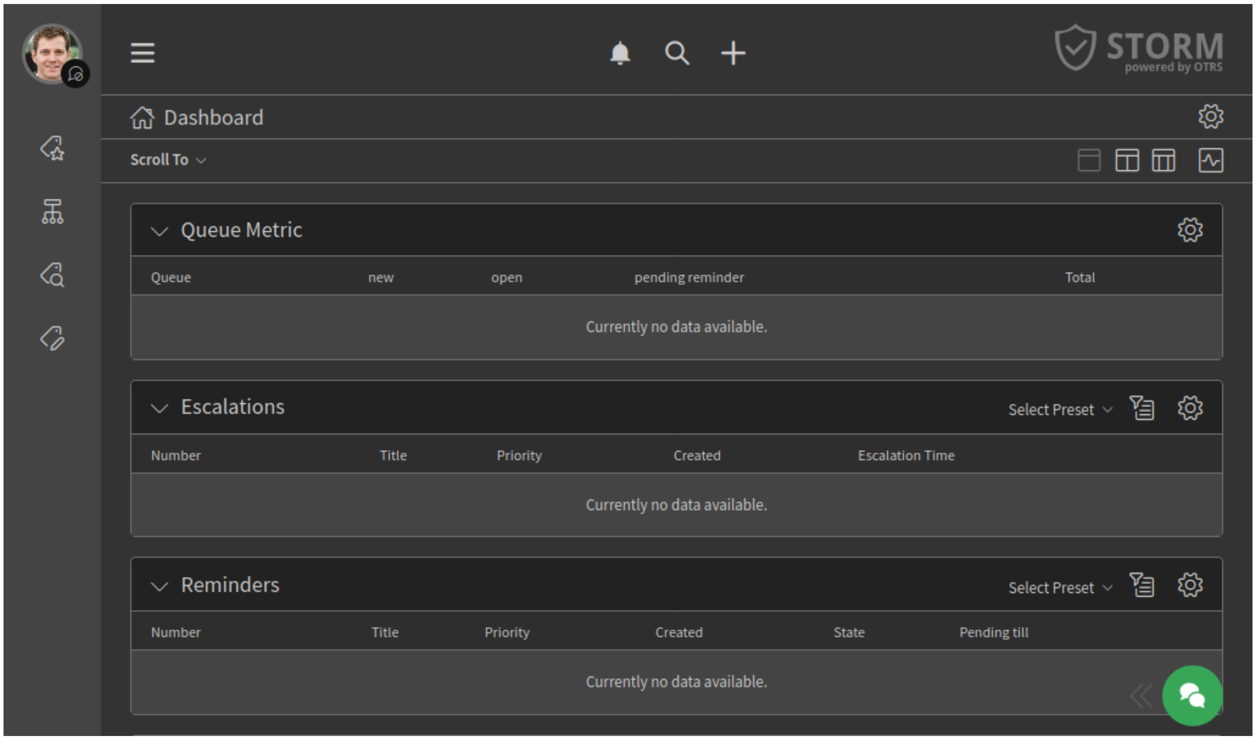
1. Go to the *Agents* module in the administrator interface.
2. Select the agent from the list of agents.
3. Click on the *Edit personal preferences for this agent* button in the left sidebar.
4. Select the *Miscellaneous* group.
5. Change the skin in the *Administrator Interface Skin* section.



Login

Please enter your credentials to log in.

Fig. 1: Login Box With Dark Theme



Queue	new	open	pending reminder	Total

Currently no data available.

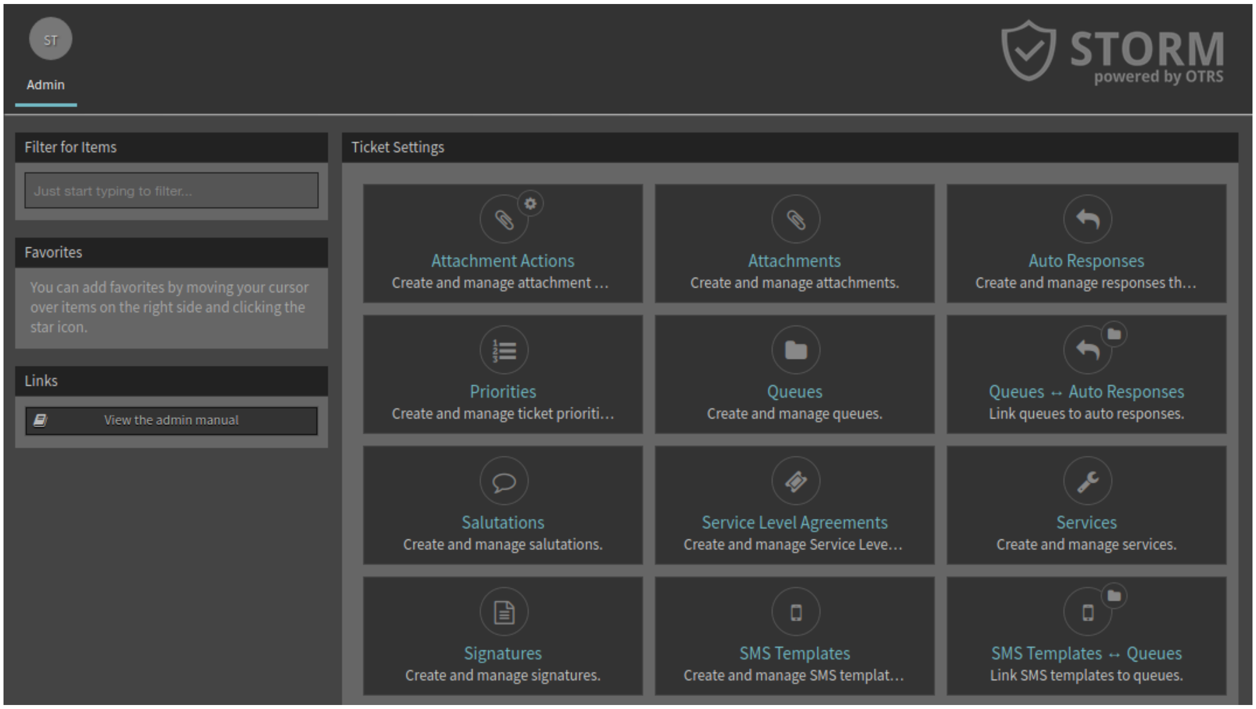
Number	Title	Priority	Created	Escalation Time

Currently no data available.

Number	Title	Priority	Created	State	Pending till

Currently no data available.

Fig. 2: Agent Interface With Dark Theme



You can add favorites by moving your cursor over items on the right side and clicking the star icon.

Attachment Actions
Create and manage attachment …

Attachments
Create and manage attachments.

Auto Responses
Create and manage responses th…

Priorities
Create and manage ticket prioriti…

Queues
Create and manage queues.

Queues ↔ Auto Responses
Link queues to auto responses.

Salutations
Create and manage salutations.

Service Level Agreements
Create and manage Service Leve…

Services
Create and manage services.

Signatures
Create and manage signatures.

SMS Templates
Create and manage SMS templat…

SMS Templates ↔ Queues
Link SMS templates to queues.

Fig. 3: Administrator Interface With Dark Skin

CHAPTER 2

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.

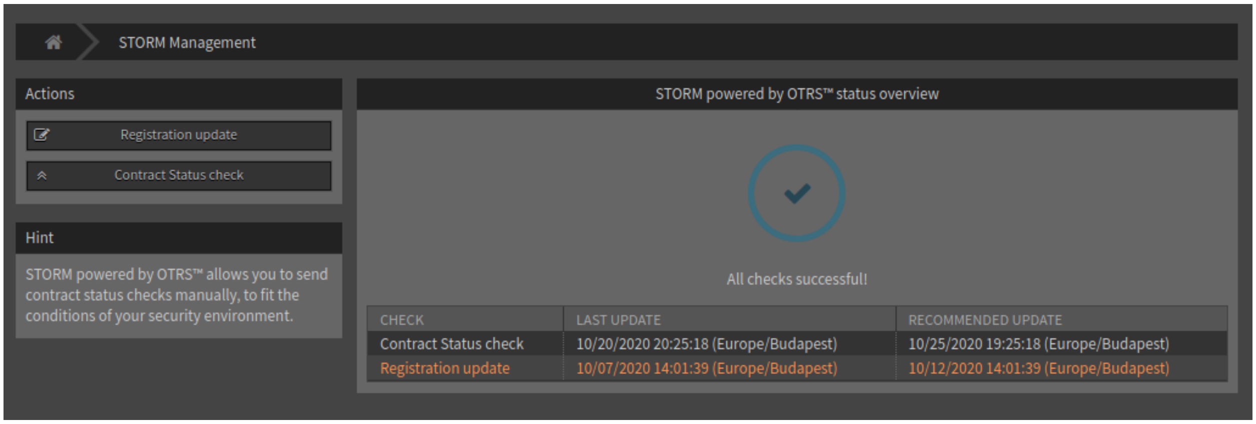
Cloud Services Automatic cloud service connection to the OTRS Group are deactivated by default. This will restrict the usage of SMS, automatic license check and registration update. To perform the needed license check an administrator has to run it manually via the *STORM Management Module*.

CHAPTER 3

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.



Hint

STORM powered by OTRS™ allows you to send contract status checks manually, to fit the conditions of your security environment.

STORM powered by OTRS™ status overview

All checks successful!

CHECK	LAST UPDATE	RECOMMENDED UPDATE
Contract Status check	10/20/2020 20:25:18 (Europe/Budapest)	10/25/2020 19:25:18 (Europe/Budapest)
Registration update	10/07/2020 14:01:39 (Europe/Budapest)	10/12/2020 14:01:39 (Europe/Budapest)

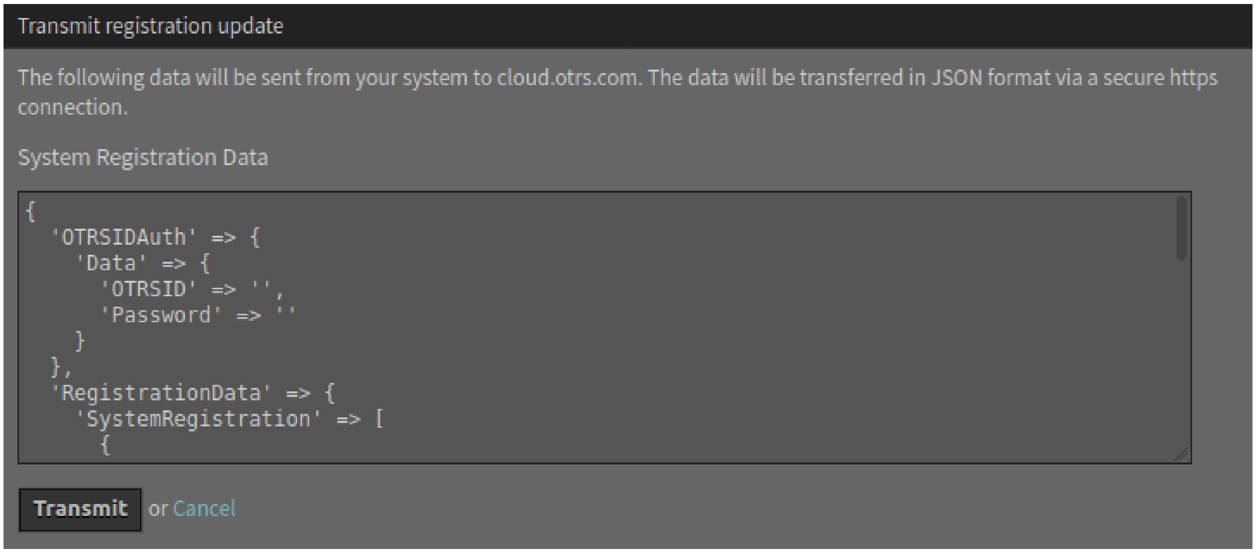
Fig. 1: STORM Management Screen

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.



Transmit registration update

The following data will be sent from your system to cloud.otrs.com. The data will be transferred in JSON format via a secure https connection.

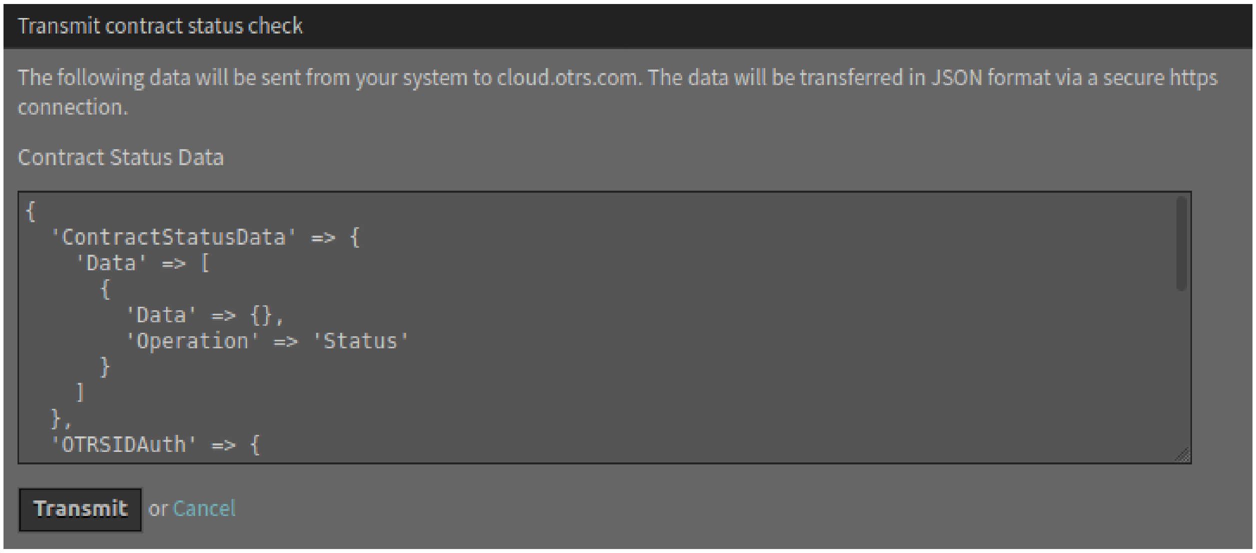
System Registration Data

```
{
  'OTRSIDAuth' => {
    'Data' => {
      'OTRSID' => '',
      'Password' => ''
    }
  },
  'RegistrationData' => {
    'SystemRegistration' => [
      {
```

Fig. 2: Registration Update Screen

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.



Transmit contract status check

The following data will be sent from your system to cloud.otrs.com. The data will be transferred in JSON format via a secure https connection.

Contract Status Data

```
{
  'ContractStatusData' => {
    'Data' => [
      {
        'Data' => {},
        'Operation' => 'Status'
      }
    ]
  },
  'OTRSIDAuth' => {
```

Fig. 3: Contract Status Check Screen

CHAPTER 4

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

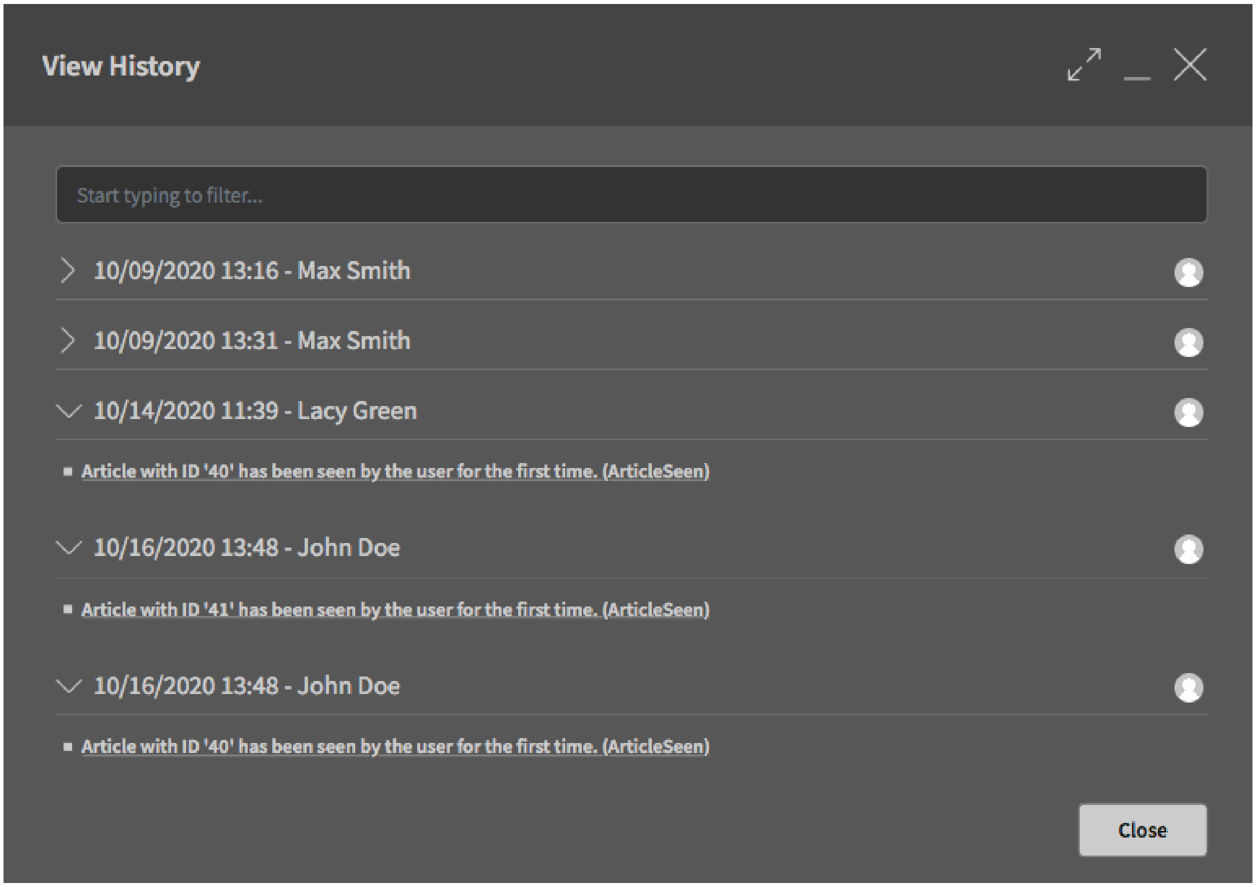
The feature adds an entry to the history, when an agent reads an article.

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 at 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.



10/09/2020 13:16 - Max Smith

10/09/2020 13:31 - Max Smith

10/14/2020 11:39 - Lacy Green

- Article with ID '40' has been seen by the user for the first time. (ArticleSeen)

10/16/2020 13:48 - John Doe

- Article with ID '41' has been seen by the user for the first time. (ArticleSeen)

10/16/2020 13:48 - John Doe

- Article with ID '40' has been seen by the user for the first time. (ArticleSeen)

Fig. 1: Example Article Seen History

CHAPTER 5

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.

5.1 Setup VirusTotal Module

The system already comes with a module to send attachments to be checked by *Virus Total* 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.

5.2 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.

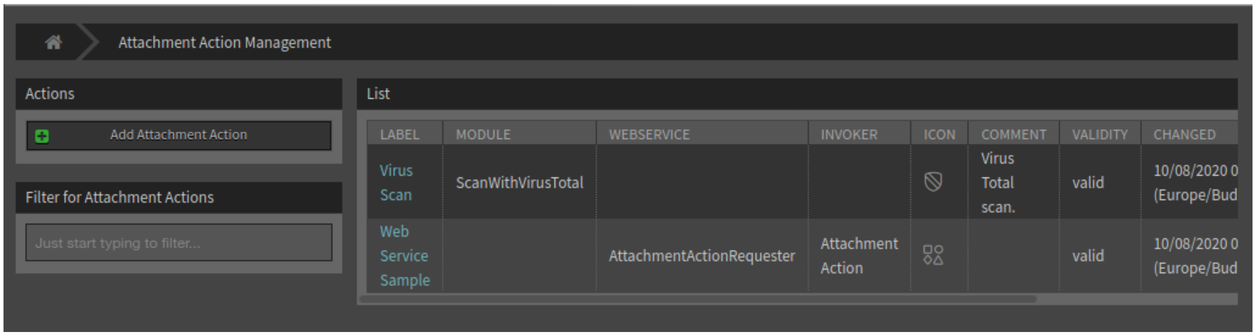
The web services can be created in the *Web Services* module of the administrator interface. The usage of this management screen is identical with the usage of the web service management screen of the OTRS framework.

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>"
```

5.3 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.



Attachment Action Management

Actions

Add Attachment Action

Filter for Attachment Actions

Just start typing to filter...

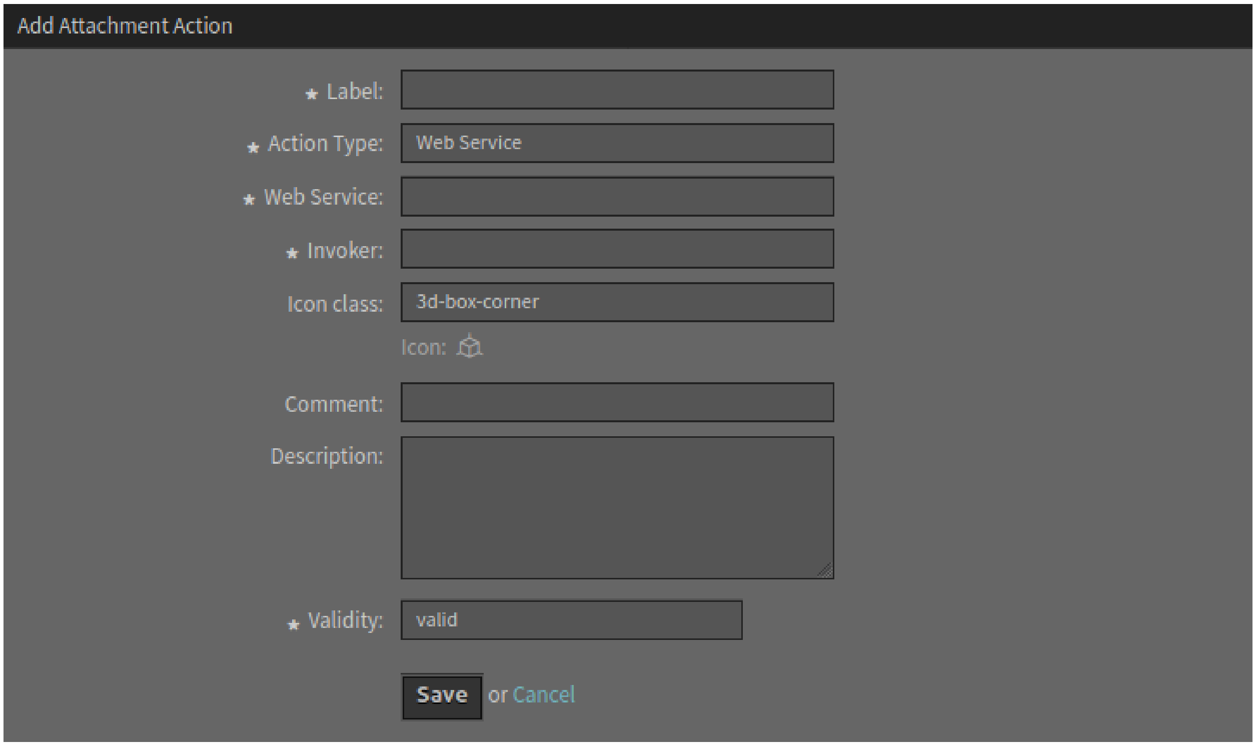
List

LABEL	MODULE	WEBSERVICE	INVOKER	ICON	COMMENT	VALIDITY	CHANGED
Virus Scan	ScanWithVirusTotal				Virus Total scan.	valid	10/08/2020 0 (Europe/Bud
Web Service Sample		AttachmentActionRequester	Attachment Action			valid	10/08/2020 0 (Europe/Bud

Fig. 1: Attachment Action Management Screen

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.



Add Attachment Action

⋆ Label:	
⋆ Action Type:	Web Service
⋆ Web Service:	
⋆ Invoker:	
Icon class:	3d-box-corner
	Icon:
Comment:	
Description:	
⋆ Validity:	valid

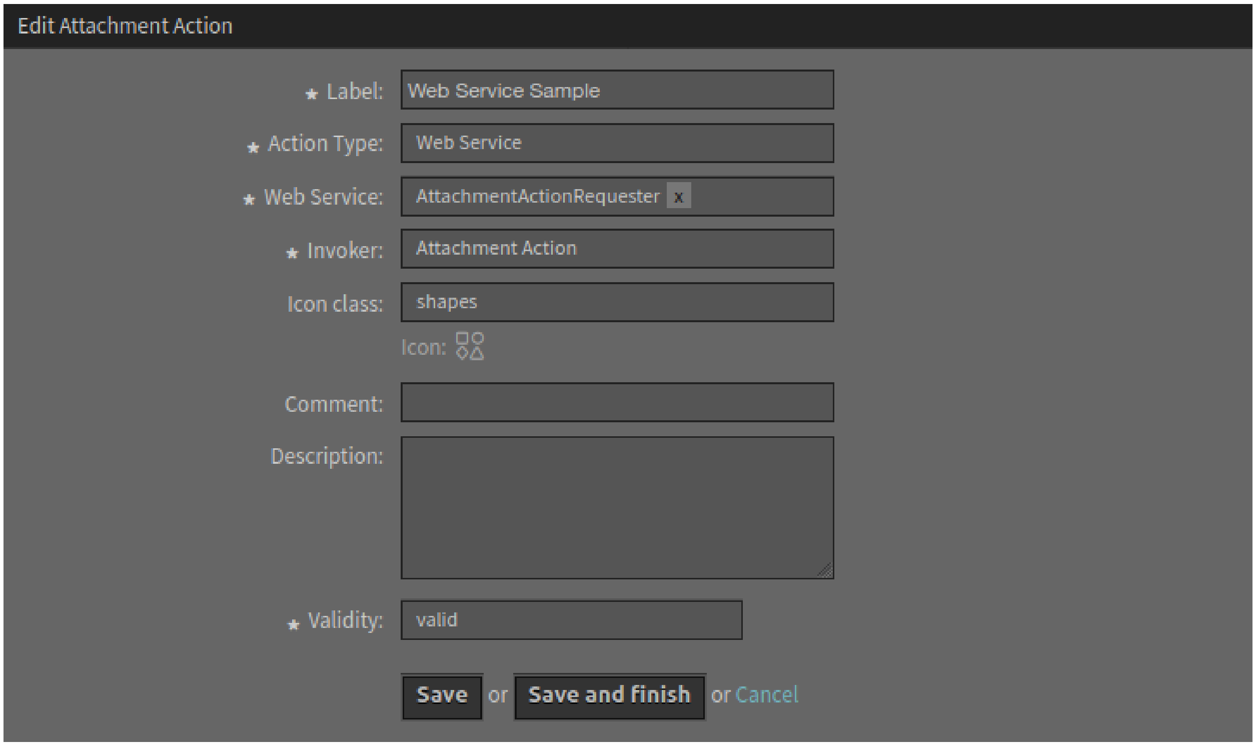
Fig. 2: Add Attachment Action Screen

It is possible to create attachment actions for modules or web services. However, only one module `ScanWithVirusTotal` is shipped with STORM, while new web services can be defined by the administrators.

> **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.



Edit Attachment Action

★ Label: Web Service Sample

★ Action Type: Web Service

★ Web Service: AttachmentActionRequester x

★ Invoker: Attachment Action

Icon class: shapes

Icon:

Comment:

Description:

★ Validity: valid

Fig. 3: Edit Attachment Action Screen

5.4 Usage

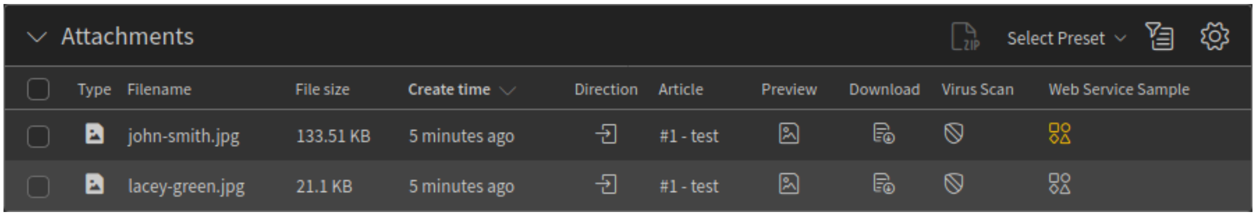
The attachment actions can be used in any attachment widget of the detail views.

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.

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.



□	Type	Filename	File size	Create time	Direction	Article	Preview	Download	Virus Scan	Web Service Sample
□		john-smith.jpg	133.51 KB	5 minutes ago		#1 - test				
□		lacey-green.jpg	21.1 KB	5 minutes ago		#1 - test				

Fig. 4: Attachments Widget

CHAPTER 6

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.

6.1 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`

6.2 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 add-on is installed, the downloads of the attachments in the dynamic fields are also logged in the system log.

CHAPTER 7

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.

7.1 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 `RegExps` 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.

7.2 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 contains 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 8

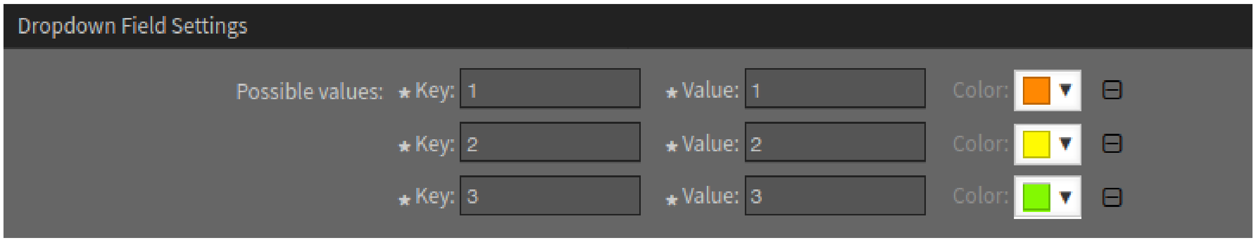
Color Indicators for Dynamic Field Values

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.



Dropdown Field Settings

Possible values: ★Key: 1 ★Value: 1 Color: 🟧 ▼ ⊟

★Key: 2 ★Value: 2 Color: 🟨 ▼ ⊟

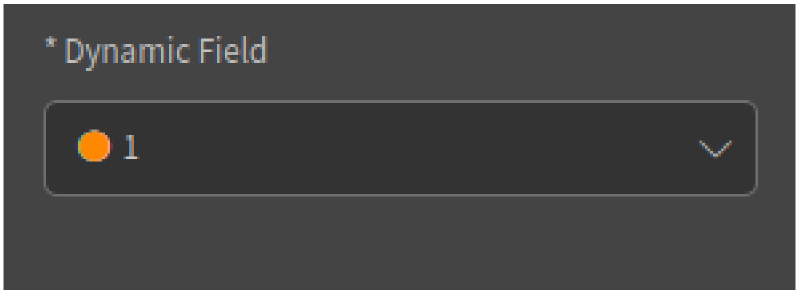
★Key: 3 ★Value: 3 Color: 🟩 ▼ ⊟

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.



* Dynamic Field

● 1

Fig. 2: Dynamic Field in Agent Interface

CHAPTER 9

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.

9.1 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.

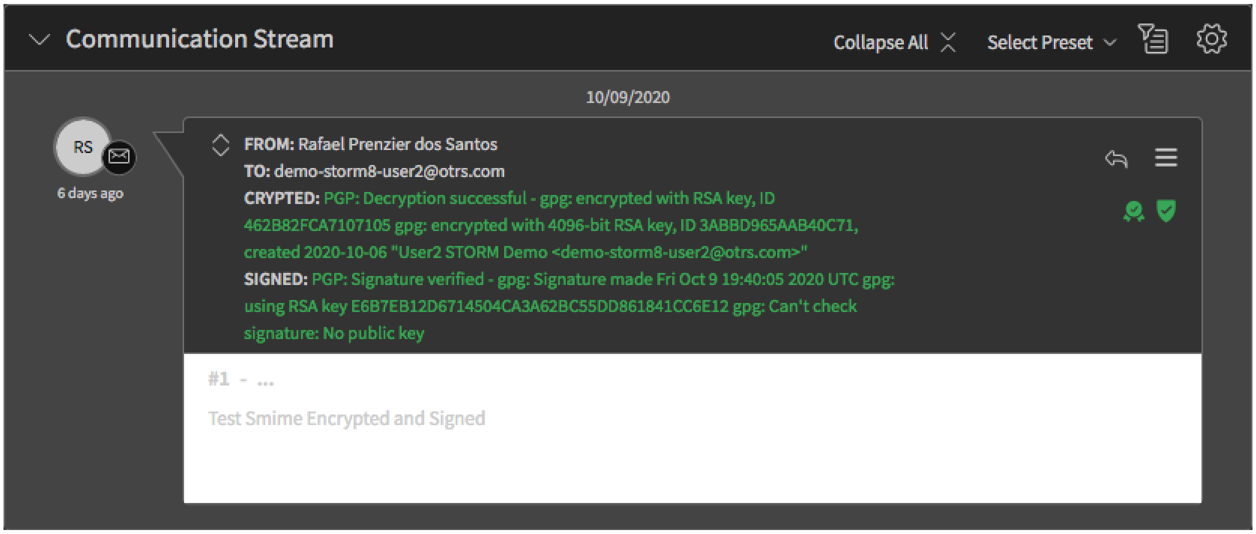
9.2 Usage

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* article action. Depending on the original message the field *Security Options* 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.



FROM: Rafael Prenzier dos Santos
TO: demo-storm8-user2@otrs.com
CRYPTED: PGP: Decryption successful - gpg: encrypted with RSA key, ID 462B82FCA7107105 gpg: encrypted with 4096-bit RSA key, ID 3ABBD965AAB40C71, created 2020-10-06 "User2 STORM Demo <demo-storm8-user2@otrs.com>"
SIGNED: PGP: Signature verified - gpg: Signature made Fri Oct 9 19:40:05 2020 UTC gpg: using RSA key E6B7EB12D6714504CA3A62BC55DD861841CC6E12 gpg: Can't check signature: No public key

#1 - ...

Test Smime Encrypted and Signed

Fig. 1: PGP Signed and Encrypted E-mail

CHAPTER 10

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.

10.1 Setup

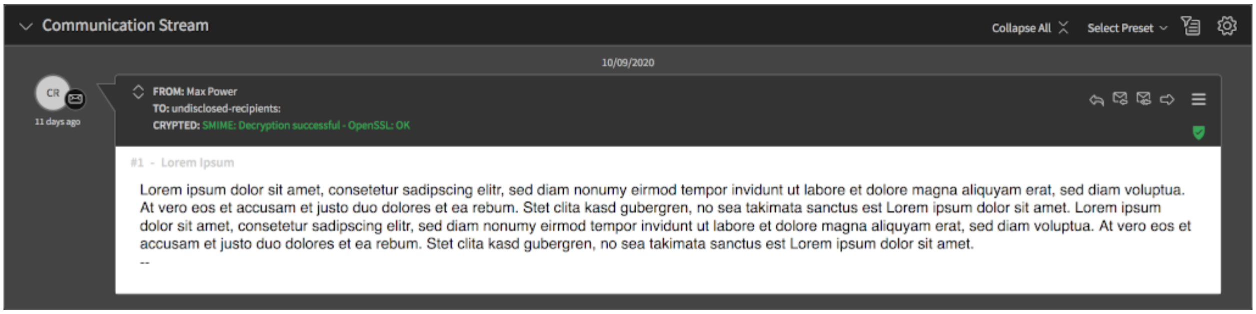
To following setup is needed for using with **S/MIME**:

- 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.

10.2 Usage

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



FROM: Max Power
TO: undisclosed-recipients:
CRYPTED: SMIME: Decryption successful - OpenSSL: OK

#1 - Lorem Ipsum

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
--

Fig. 1: Decrypted Bcc Email Example

CHAPTER 11

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.

11.1 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`

11.2 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'.
```

(continues on next page)

(continued from previous page)

CHAPTER 12

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.

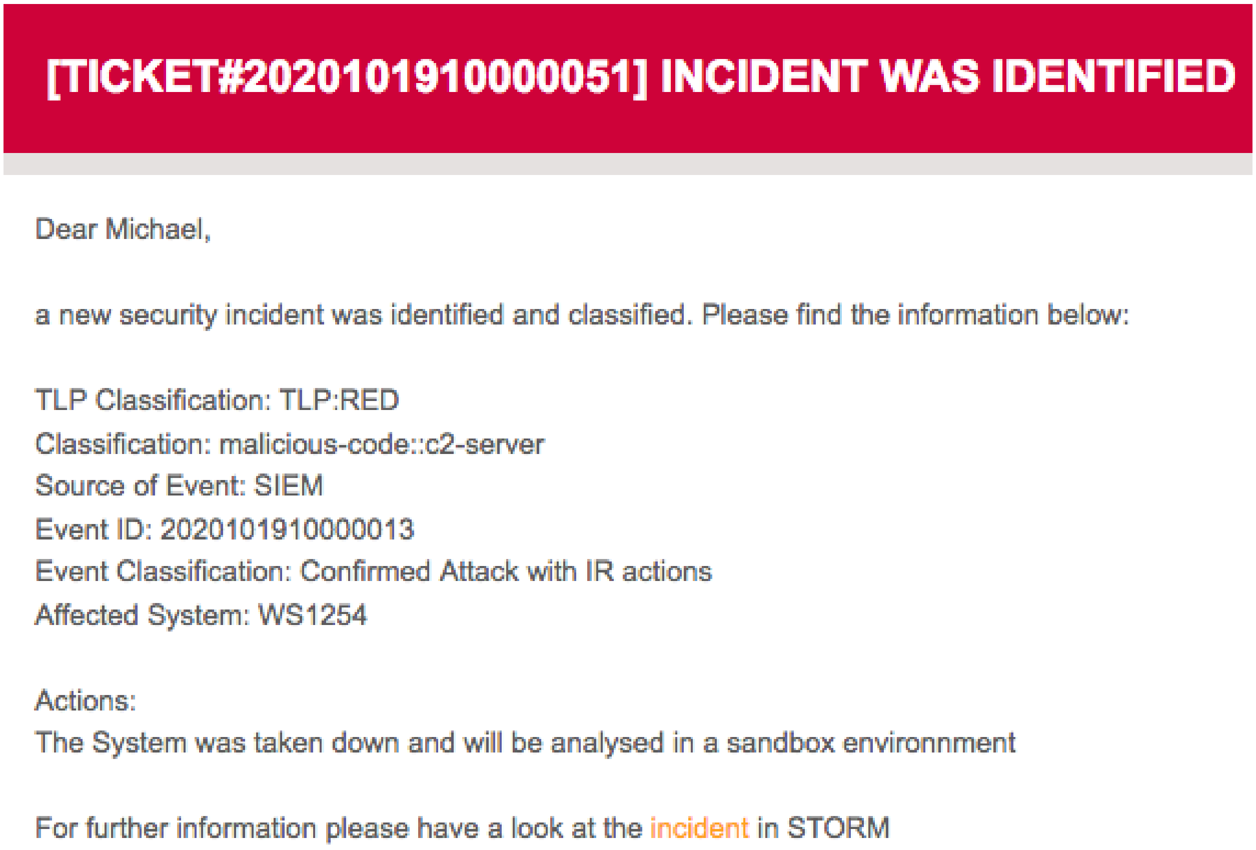
STORM comes with four pre-designed templates:

- `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 13

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 uses 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.

13.1 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*.

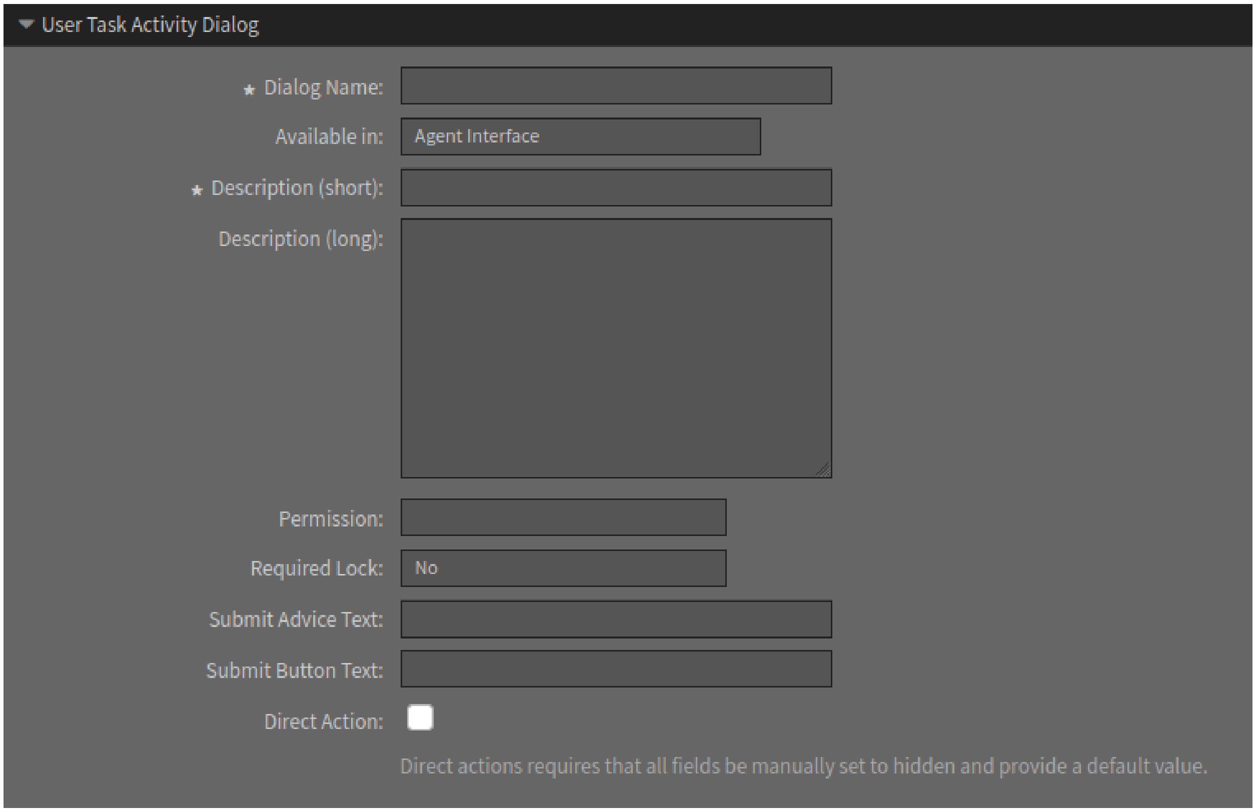
The user task activity dialog is extended with a new field *Direct Action*. If this field is checked, the activity dialog will be submitted automatically.

Direct actions requires that all fields 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.

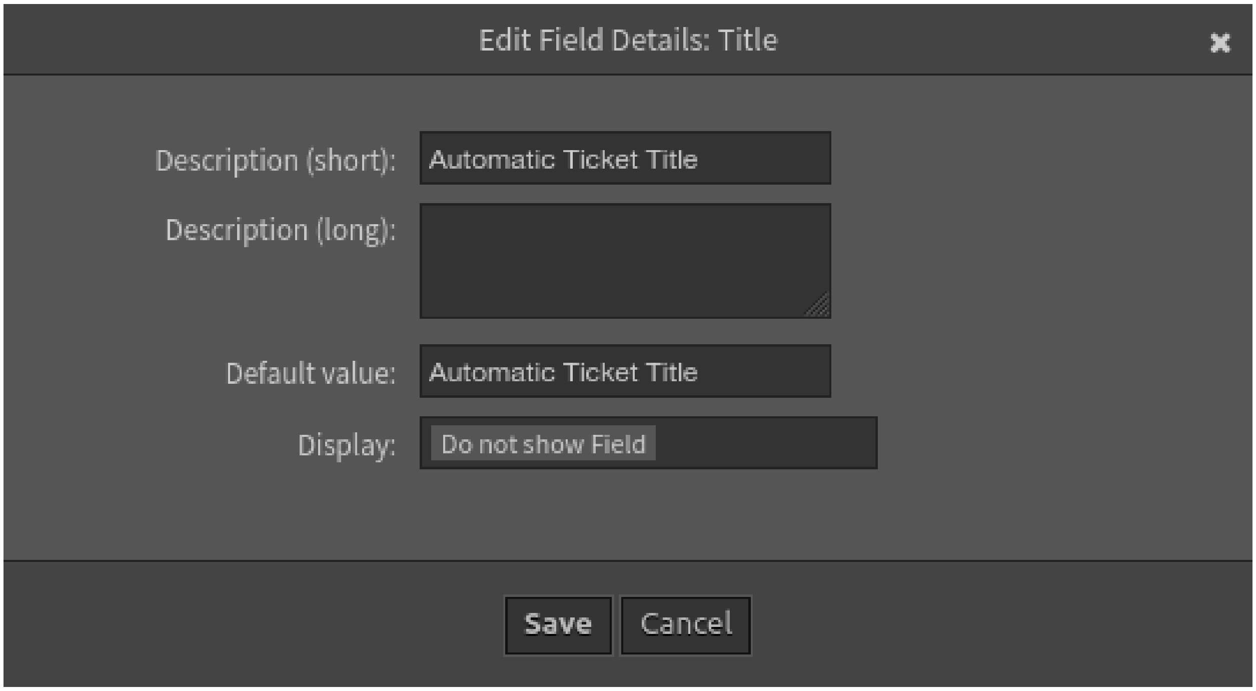


User Task Activity Dialog

★ Dialog Name:	
Available in:	Agent Interface
★ Description (short):	
Description (long):	
Permission:	
Required Lock:	No
Submit Advice Text:	
Submit Button Text:	
Direct Action:	□

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

Fig. 1: Edit User Task Activity Dialog Window



Edit Field Details: Title

Description (short):	Automatic Ticket Title
Description (long):	
Default value:	Automatic Ticket Title
Display:	Do not show Field

Fig. 2: Edit Field Details Dialog

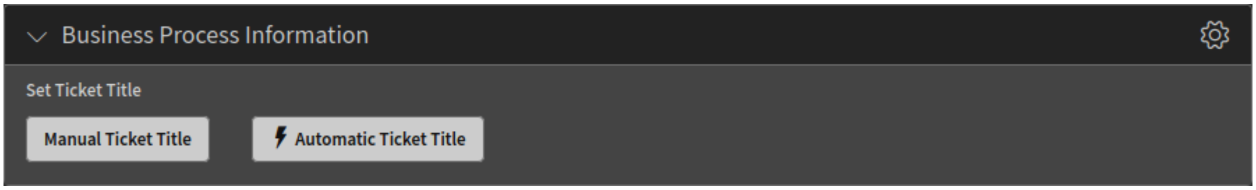


Fig. 3: Edit Field Details Dialog

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.

CHAPTER 14

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

The result values can been access from:

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

<Output> Any text produced to the standard output.

<ErrorOutput> Any text produces 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.

14.1 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`.