



SelfCare SDK
for



Version 4.5
Part Number E62400-01

Table of Content

1 Introduction.....	4
1.1 Document Purpose.....	4
1.2 Scope of the Document.....	4
1.3 Target Audience.....	4
1.4 Glossary.....	4
2 SelfCare Overview.....	6
2.1 Goals and Objectives.....	6
2.2 Security Issues.....	6
2.3 SelfCare Implementation.....	6
2.4 Usage Examples.....	6
3 Common SelfCare Entities and Structures.....	7
3.1 Properties and Fields.....	7
3.1.1 File Properties.....	7
3.1.2 Property Visibility.....	7
3.1.3 'properties' Structure.....	8
3.2 Resources.....	8
3.2.1 Resource Type.....	8
3.2.2 Resource Properties.....	8
3.3 Activities.....	9
3.3.1 Activity Types.....	9
3.3.2 Activity Timing Details.....	9
3.3.3 Activity Statuses.....	10
3.3.4 Activity Properties.....	10
3.4 Transaction Authentication.....	14
3.4.1 'user' Authentication Structure.....	14
3.4.2 Authentication.....	14
4 Detailed Methods Description.....	15
4.1 'search_activities' Method.....	15
4.1.1 'search_activities' Request.....	15
4.1.2 'search_activities' Response.....	17
4.2 'update_activity' Method.....	18
4.2.1 'update_activity' Request.....	18
4.2.2 'update_activity' Response.....	19
4.3 'get_file' Method.....	22
4.3.1 'get_file' Request.....	22
4.3.2 'get_file' Response.....	23
4.4 'get_resource' Method.....	24
4.4.1 'get_resource' Request.....	24
4.4.2 'get_resource' Response.....	24
5 Transaction Errors.....	27
5.1 SOAP Faults.....	27
5.2 Error Responses.....	28

This document contains proprietary and confidential information of TOA Technologies and shall not be reproduced or transferred to other documents, disclosed to others, or used for any other purpose other than that for which it is furnished, without the prior written consent of TOA Technologies. It shall be returned to TOA Technologies upon request. The trademark and logo of TOA Technologies are the exclusive property of TOA Technologies, and may not be used without permission. All other marks mentioned in this material are the property of their respective owners.

5.3 Error Codes.....	29
5.3.1 Activity Management API Errors.....	29
5.3.2 Resource Management API Errors.....	29
6 Migration from 4.4 to 4.5.....	30
6.1 'user' Authentication Structure.....	30

1 Introduction

1.1 Document Purpose

The document is meant to ensure successful interaction of Client-developed applications and the ETAdirect application server, when they are related to SelfCare applications.

1.2 Scope of the Document

The document provides general description of the SelfCare functionality and relevant SOAP elements as well as the methods used therewith.

1.3 Target Audience

This document is intended for developers of SOAP client applications.

1.4 Glossary

Term	Explanation
Activity	Entity of the ETAdirect system that represents any time-consuming activity of the resource
Activity Status	Dynamic value that corresponds to the state of a particular activity execution
API	Application Programming Interface – a particular set of rules and specifications that software programs follow to communicate and interact with each other
Bucket	Entity appearing on the resource tree which can contain resources of a defined type and be assigned activities
Client Application	See SOAP Client Application
Customer	End-Customer, entity that benefits from the Activity
ETA	Predicted time at which a resource will arrive at an appointment and start an activity, calculated dynamically from current and historical data
ISO 8601 format	See http://en.wikipedia.org/wiki/ISO_8601
Property	Field and field value assigned to an entity in ETAdirect (to user, resource, activity or inventory). There are fields and custom properties
Resource	Element in the resource tree representing a defined company asset
Resource External ID	Company-unique key used to identify a specific resource
Route	List of activities assigned to a resource for a specific date, or a list of non-scheduled activities assigned to a resource
SOAP	Lightweight protocol for exchange of information in a decentralized, distributed environment
SOAP 1.1	See http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
SOAP Interface	Interface used to receive requests and return responses via SOAP
SOAP Client Application	Application running at the Client's site and providing interaction with ETAdirect server via SOAP
SOAP Fault	SOAP element used to carry error and/or status information in a SOAP message
Technician (Resource)	Person who performs work at the customer's premises (the one who provides the service)
User	1) Person using ETAdirect

Term	Explanation
	2) Entity used for authentication and authorization, allowing people or external software to access ETAdirect

2 SelfCare Overview

2.1 Goals and Objectives

SelfCare functionality is used to support the integration of ETAdirect with the corporate web-site of the ETAdirect client company that renders services to a customer. This enables the client, the company's customer, or a third party, to track services (deliveries) the customer ordered from the company (get and change the activity information for a specific customer based on the service identification and/or contact information). On the other hand, SelfCare supports the ability to retrieve the data of the specific resource, such as name, contact or calendar details, etc.

2.2 Security Issues

The security is ensured, as the company's administrator can only delegate to the user, or a third party, SOAP interface access rights which he/she has obtained from TOA Technologies.

2.3 SelfCare Implementation

In order to implement SelfCare functionality, the following methods of Activity Management API and Resource Management API (optional) need to be used.

Activity Management API methods that can be used for SelfCare:

- 'search_activities' – extracts activity information
- 'update_activity' – changes activity information
- 'get_file' – gets file properties of an activity or a technician.

Resource Management API methods that can be used for SelfCare:

- 'get_resource' – checks whether an activity is assigned to a technician, determines whether the activity is assigned to an individual technician or to a bucket, gets the resource's information such as personal name.

Note: ETAdirect still supports the obsolete SelfCare API for the existing customers, but it is not recommended that the obsolete API is used for new development.

(The obsolete SelfCare API had 'get_customer_eta', 'set_customer' methods).

2.4 Usage Examples

One of the major functions of the functionality is to let the end-user get an answer to 'Where is my technician?' question. The answer is easy to find with the 'search_activity' method. The method enables transmitting to ETAdirect the data on the time period to search for activity in, the activity details and its specific search value (e.g. name of the customer) and preferences as to the way the search results are returned (max amount, order, details of the activity found). So the Client may log in to the Company Web portal and enter the name and get details of the task time, engineer, etc. It is possible to limit the set of details that can be returned.

Another way to use the functionality is to let the users change details of an activity (e.g. if they cannot be present at the time initially set). The 'update_activity' method can be used for that purpose. In order to update the activity the user has to enter the activity identifier. The identifier can be provided to the user initially (as it is done in SelfCare portals of mail delivery tracking) or may be returned with the 'search_activities' method (e.g. if the user forgot the number). This provides additional security and

ensures that only authorized users can change activity data.

Another thing that may be useful for SelfCare purpose is the user's ability to get file properties – this may be the photo of the technician who is to perform the task, certain agreements, as well as other documentation. This is possible with the 'get_file' method. In order to use the method it is necessary to know the ID of the activity or the external ID of the resource to which the property is assigned and the label of such property (label of the resource's photo or of the agreement). The activity ID or the resource external ID may be retrieved with the 'search_activities' method.

The optional 'get_resource' method allows the user to get the data of the resource, for example, their phone number or email address to be able to contact the resource in case of any changes in the agreed procedure. To use the 'get_resource' method the user needs to know the external ID of the resource. In its turn, this external ID can be returned by the 'search_activities' method.

3 Common SelfCare Entities and Structures

3.1 Properties and Fields

Property is a variable associated with some ETAdirect entity. This is much like a field of an object in most programming languages. Every property has a string label by which it is uniquely identified – it can be thought of as a field name and a value.

SelfCare functionality processes properties of activities and/or resources in ETAdirect. Some properties are initially defined in ETAdirect and others are client-specific and created in the course of implementation, though as soon as there is an activity property in ETAdirect, all activities have this property, and SelfCare can read this property, write to it, and make conditional decisions based on the property value. Similarly, a property defined for a resource type will be assigned to all resources of the same type in ETAdirect. However, SelfCare only supports retrieval of resource data without an ability to change it.

Properties initially defined in ETAdirect are addressed as 'fields'.

3.1.1 File Properties

If a file is assigned to an entity, it is called a file property or file. File data is always encoded using base64 encoding.

The SelfCare functionality allows to retrieve file properties of the specified entities. This is achieved by means of the 'get_file' method. To retrieve a file property the user must send the ID of the entity for which the file property has to be retrieved and the label of the file property itself.

3.1.2 Property Visibility

The way a property will be processed by SelfCare depends on its visibility. A property can be set to hidden and will not be seen in any way by the user. On the other hand, visible properties can be mandatory for the request to be processed correctly or optional.

Optional: the user can see the property and can optionally manage it. The 'Required' column contains '**No**' for such property.

- Mandatory:**
- the user can see the property and must define it
 - if the transaction contains an invalid mandatory property, the request is rejected with a corresponding error
 - if request has no mandatory property, the request is rejected with a corresponding error

The 'Required' column contains '**Yes**' for such properties.

3.1.3 'properties' Structure

The 'properties' structure represents an activity or resource property in the form of a simple name-value pair consisting of the following elements:

Name	Required	Type	Description
name	Yes	string	property name, unique for the corresponding property list
value	Yes	string	property value, can be an empty string. When an empty string is sent, the value of the existing property is deleted
mime_type	No	string	used only for file properties in the 'get_resource' response
file_data	No	base64 binary	used only for file properties in the 'get_resource' response

3.2 Resources

Resource is an entity that represents a person, group of persons or equipment used by a company, so that it or its child resources can perform work for the benefit of a company. For each resource entity-specific parameters are defined.

3.2.1 Resource Type

Resource Type contains a predefined set of company-specific rules for a resource (e.g. the ability of the resource to have child resources, to be assigned and execute activities, to assist and be assisted by other resources, share data with other resources and interact with different modules of ETAdirect, etc.).

3.2.2 Resource Properties

In addition to the fields listed in the table below, custom resource properties defined in the specific company can also be returned.

Name	Label	Type	Description
id	external_id	string	resource external ID
status	pactive	enum	resource status (active or inactive)
parent_id	-	string	parent resource external ID for the top resource it is equal to 0
type	ptype	enum	resource type company specific value
name	pname	string	resource name
language	planguage	enum	resource language company specific language label (EN, FR, etc.)
email	email	string	resource email
phone	pphone	string	resource phone
time_zone	time_zone	enum	resource time zone

Name	Label	Type	Description
			company specific time zone name
time_from	-	time	HH:MM format start of resource's working hours according to the calendar for the specified date (is returned only if the optional valid 'date' field is present in the request and the resource has a working day)
time_to	-	time	HH:MM format end of resource's working hours according to the calendar for the specified date (is returned only if the optional valid 'date' field is present in the request and the resource has a working day)
non_working_reason	-	string	if the 'date' field is present in the request and the resource has a non-working day on the specified date, the reason will be returned in this field. The reason code is one of the configurable list of constants, e.g. "Vacation"
date_format	pdate_fid	enum	date format set for the resource
time_format	ptime_fid	enum	time format set for the resource
workskill	resource_workskills	string	work skill defined in ETAdirect and acquired by the resource
workskill_group	-	string	work skill group defined in ETAdirect and acquired by the resource

3.3 Activities

Activities are entities that correspond to time-consuming work performed for the benefit of the company. Upon creation a set of parameters (properties) is defined for the activity. The accessibility and mandatory/optional requirements of the properties are defined with the user's settings. The properties defined for an activity include:

IDs: automatically generated numeric ID

Customer and activity details: address, phone number, name, language, activity site, special preferences that define to what resource the activity can be assigned.

Activity type: one of the predefined company-specific [activity types](#)

Other: any custom properties required

3.3.1 Activity Types

An activity type corresponds to a predefined company-specific set of rules applied when processing an activity. The rules cover the resources activity can be assigned to, details of its processing and interaction with different modules of ETAdirect (e.g. statistics, notification).

The 'Allow to search' feature must be enabled for the activity to be available for search ('Manage Application' → 'Company Settings' → 'Activity Types').

3.3.2 Activity Timing Details

When an activity is modified its timing detail can also be defined. They can include:

date – specific date when activity has to be performed – date of the route to which the activity is assigned. **Note:** date cannot be updated by the SelfCare functionality.

SLA window – a date range within which the activity has to be performed (started after the start and complete before the end). Usually SLA is the time agreed with the customer for the activity performance

service window – a period of time during the day within which the activity has to be started

- usually it is the time agreed with the customer for the activity performance start
- can be updated before the activity start

If both SLA window and service window are defined for the activity, it should be performed within their overlap period.

time slot – company-specific labeled service windows which can be defined in the system and referred to (e.g. time slot 'Lunch' = service window 12 p.m. – 1 p.m.)

3.3.3 Activity Statuses

As the resource performs the activity it changes its status. Using the activity status, ETAdirect can define the stage of the activity performance and initiate company specific notifications and reports.

Pending

- when activity is created in the resource's route it gets 'pending' status

Cancelled

- activity that has not been started and will not be performed is of 'cancelled' status

Started

- activity that has been started and is being processed is of 'started' status

Suspended

- setting started activity temporary ended and creating duplicating not-ordered pending activity

Complete

- activity that has been successfully completed is of 'complete' status

Not done

- activity that has been started but has not been completed is of 'notdone' status

Reopened

- activity that was set complete, cancelled or not done can be reopened
- new not-ordered pending activity will be created

Pework: In addition to regular (initial and reopened) activities, there is prework in ETAdirect, processed with the Activity Management API. Pework is work necessary to perform a specific activity, it is always created in the 'started' status and can be completed or delayed.

3.3.4 Activity Properties

In addition to the fields listed in the table below, custom activity properties defined in the specific company can also be returned.

Name	Label	Type	Value	Update Allowed
id	aid	int	activity ID	No
appt_number	appt_number	string	work order number	Yes
resource_id	-	string	resource external ID	No
type	atype	enum	whether the activity is regular or pre-work	No

Name	Label	Type	Value	Update Allowed
status	astatus	enum	activity status: pending; started; suspended; canceled; deleted; complete; notdone	No
worktype	aworktype	enum	activity work type	No
workzone	aworkzone	enum	activity work zone	No
duration	length	int	length (duration) of the activity in minutes	Not allowed for work types with 'predefined duration' feature enabled, otherwise allowed
time_slot	time_slot	enum	activity time slot (string label) available only if time slots are defined for the company ('Manage Application' → 'Company Settings' → Time Slots)	Yes
service_window_start	service_window	time	customer service window start time in (H)H:MM format (e.g 8:15, 08:15, and 14:30)	Yes
service_window_end	service_window	time	customer service window end time in (H)H:MM format	Yes
delivery_window_start	delivery_window	time	activity delivery window start in (H)H:MM format	Yes
delivery_window_end	delivery_window	time	activity delivery window end in (H)H:MM format	Yes
sla_window_start	sla_window_start	DateTime	activity SLA window start in YYYY-MM-DD HH:MM:SS format	Yes
sla_window_end	sla_window_end	DateTime	activity SLA window end in YYYY-MM-DD HH:MM:SS format	Yes
name	cname	string	customer's name	Yes
customer_number	customer_number	string	customer's account number	Yes
phone	cphone	string	customer's regular (land) phone number	Yes
email	cemail	string	customer's email address	Yes
cell	ccell	string	customer's cell phone number	Yes

Name	Label	Type	Value	Update Allowed
address	caddress	string	customer's address Note: this field is used by geocoding and, therefore, must contain a valid address. Other values will not be resolved correctly by the geocoding server.	Yes
city	ccity	string	customer's city of residence Note: this field is used by geocoding and, therefore, must contain a valid city name. Other values will not be resolved correctly by the geocoding server.	Yes
zip	czip	string	customer's zip/post code Note: this field is used by geocoding and, therefore, must contain a valid zip/post code. Other values will not be resolved correctly by the geocoding server.	Yes
state	cstate	string	customer's state of residence Note: this field is used by geocoding and, therefore, must contain a valid state name. Other values will not be resolved correctly by the geocoding server.	Yes
language	clanguage	enum	notification language company specific language label (EN, FR, etc.)	Yes
reminder_time	cmessagetime	int	reminder notification time: how many minutes before the activity start time the customer should be notified	Yes
time_zone	c_zid	enum	customer's time zone	Yes
coord_status	acoord_status	string	whether or not activity coordinates were found	No
coordx	acoord_x	float	latitude of the activity (of the customer's location)	Yes
coordy	acoord_y	float	longitude of the activity (of the customer's location)	Yes
start_time	ETA	DateTime	ETA time (for started and ended activities – time when the activity was started) in YYYY-MM-DD HH:MM:SS format	No
end_time	end_time	DateTime	predicted or actual end time of activity in YYYY-MM-DD HH:MM:SS format	No
date	date	date	activity date in YYYY-MM-DD format	No

Name	Label	Type	Value	Update Allowed
team_id	-	string	external ID of the team-holder – the head resource within a team	No
unordered	-	enum	returned in the response with value = '1' if there is no specific time within the resource's route when the activity has to be performed (e.g. to perform first, last, after the first etc.)	No
position_in_route	position_in_route	int	if there is a specific time within the resource's route when the activity has to be performed, 'position_in_route' is the number of the activity in the route	No

3.4 Transaction Authentication

3.4.1 'user' Authentication Structure

All SelfCare methods use the 'user' structure as request authentication in order to determine the permissions of the ETAdirect client company user. The ETAdirect SOAP authentication structure has the following **mandatory** fields:

Name	Type	Description
now	string	current time in ISO 8601 format
company	string	case-insensitive identifier of the Client for which data is to be retrieved provided by TOA Technologies during integration
login	string	case-insensitive identifier of a specific user within the Company provided by TOA Technologies during integration
auth_string	string	authentication hash; auth_string = md5(now + md5(password)); where 'password' is a case-sensitive set of characters used for user authentication provided by TOA Technologies during integration

For example:

```
<user>
  <now>2014-01-10T13:56:50Z</now>
  <login>soap</login>
  <company>in132</company>
  <auth_string>ba8e6417c60e0e1748265252b4f4d601</auth_string>
</user>
```

3.4.2 Authentication

The 'user' structure is used for the request authentication. If any of the situations below occur, authentication fails and the relevant error is returned.

Authentication fails if:

- 1 now is different from the current time on the server and this difference exceeds the predefined time-window (30 minutes by default)
- 2 company cannot be found in the ETAdirect
- 3 login cannot be found for this company
- 4 user with this 'login' is not authorized to use the current method
- 5 auth_string is not equal to md5(now+md5(password))

For example:

'now' = "2005-07-07T09:25:02+00:00" and password = "Pa\$\$w0rD"
then
md5 (password) = "06395148c998f3388e87f222bfd5c84b"
concatenated string =
= "2005-0707T09:25:02+00:0006395148c998f3388e87f222bfd5c84b"

auth_string should be:

auth_string = "62469089f554d7a38bacd9be3f29a989"

Otherwise authentication is successful and the request is processed further.

4 Detailed Methods Description

4.1 'search_activities' Method

The 'search_activities' method enables retrieving the list of activities with the specified value in the specified field for the specified time period.

4.1.1 'search_activities' Request

The 'search_activities' method request specifies:

- time period in which to search for an activity and the activity property value to search for
- the way activities should be ordered in the response
- properties to be retrieved for the activity specified

The 'search_activities' method request contains the following elements:

Field	Required	Type	Description
user	Yes	struct	' user ' structure
search_in	Yes	string	activity field name or custom property label in which to search for 'search_for' value. The following properties are available for search: <ul style="list-style-type: none"> • address • ccell • cphone • cemail • cname • appt_number • customer_number • czip • labels of customer properties. Note: ONLY PROPERTIES DEFINED IN 'Manage Application' → 'Company Settings' → 'Business Rules' → Search Fields can be used
search_for	Yes	string	value to search for
date_from	Yes	date	beginning of the date interval for the search in the YYYY-MM-DD format
date_to	Yes	date	end of the date interval for the search n the YYYY-MM-DD format
select_from	Yes	int	specifies area of the list of activities found to return data from; activities from the route starting with 'select_from' are returned
select_count	Yes	int	total number of activities to be returned
order	No	string	enum value defines the order of the activities found; valid values: asc – ascending; desc – descending; by default is set to descending
property_filter	No	array	array of ' properties ' names or custom property labels, each containing one of activity properties to be returned for activities found if omitted, 'id', 'resource_id' and 'date' will be returned for all found activities, where available

- **'search_activities' Request Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:search_activities>
      <user>
        <now>2014-01-10T13:56:50Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>ba8e6417c60e0e1748265252b4f4d601</auth_string>
      </user>
      <search_in>appt_number</search_in>
      <search_for>test_sea</search_for>
      <date_from>2014-01-09</date_from>
      <date_to>2014-01-11</date_to>
      <select_from>1</select_from>
      <select_count>1000</select_count>
      <order>desc</order>
      <property_filter>date</property_filter>
    </ns1:search_activities>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


4.1.2 'search_activities' Response

The response to the 'search_activities' request contains data on the method success/failure and activity properties specified in the request.

Field	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error_description (if 'result_code' is other than '0')
activity_list	struct	properties specified for the activities found
activity_list/total	int	number of the activities found
activity_list/activities	array	list of activities found and properties specified in the 'property_filter' if 'property_filter' is empty, 'id', 'resource_id' and 'date' will be returned for all found activities, where available

- 'search_activities' Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:search_activities_response>
      <result_code>0</result_code>
      <activity_list>
        <total>2</total>
        <activities>
          <activity>
            <properties>
              <name>date</name>
              <value>2014-01-10</value>
            </properties>
          </activity>
          <activity>
            <properties>
              <name>date</name>
              <value>2014-01-10</value>
            </properties>
          </activity>
        </activities>
      </activity_list>
    </ns1:search_activities_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.2 'update_activity' Method

The 'update_activity' method is used to alter the specified activity properties in ETAdirect.

4.2.1 'update_activity' Request

The 'update_activity' method request specifies:

- activity to be updated
- properties to be updated for the specified activity
- position in the route of the specified activity

The 'update_activity' method request contains the following elements:

Field	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity to be updated
position_in_route	Yes	string	ID of the activity that will be followed by the processed activity valid values: any company specific activity ID special keys: 'unchanged': activity order is not changed 'unordered': processed activity is not-ordered 'first': processed activity is the first 'last': processed activity is the last
properties	No	array	array of 'properties' nodes, each containing some activity properties and/or custom properties to be set for the updated activity if an empty 'value' is sent for a property, the existing value is deleted all properties are optional for the method

- **'update_activity' Request Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:update_activity>
      <user>
        <now>2014-01-10T14:05:21Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>b029ee847c113e8f3ec0a8c916c29c87</auth_string>
      </user>
      <activity_id>3998006</activity_id>
      <position_in_route>unchanged</position_in_route>
      <properties>
        <name>phone</name>
        <value>0066744031</value>
      </properties>
    </ns1:update_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.2.2 'update_activity' Response

The response to the 'update_activity' request contains data on the method success/failure and all properties of the processed activity.

The response of the 'update_activity' method contains the following elements:

Field	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error_description (if 'result_code' is other than '0')
activity	array	array of properties nodes, each containing one of activity_properties and/or custom properties for the updated activity all available activity properties are returned

- **'update_activity' Response Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:update_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <name>name</name>
          <value>Mister Roboto</value>
        </properties>
        <properties>
          <name>zip</name>
          <value>12345</value>
        </properties>
        <properties>
          <name>phone</name>
          <value>0066744031</value>
        </properties>
        <properties>
          <name>time_zone</name>
          <value>Eastern</value>
        </properties>
        <properties>
          <name>type</name>
          <value>regular</value>
        </properties>
        <properties>
          <name>position_in_route</name>
          <value>1</value>
        </properties>
      </activity>
    </ns1:update_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<properties>
  <name>worktype</name>
  <value>33</value>
</properties>
<properties>
  <name>time_slot</name>
  <value>08-10</value>
</properties>
<properties>
  <name>service_window_start</name>
  <value>14:00:00</value>
</properties>
<properties>
  <name>service_window_end</name>
  <value>16:00:00</value>
</properties>
<properties>
  <name>appt_number</name>
  <value>test_update_appointm-1-ZKSEDLSCBQ</value>
</properties>
<properties>
  <name>language</name>
  <value>en</value>
</properties>
<properties>
  <name>duration</name>
  <value>48</value>
</properties>
<properties>
  <name>start_time</name>
  <value>2014-01-10 14:00:00</value>
</properties>
<properties>
  <name>status</name>
  <value>pending</value>
</properties>
<properties>
  <name>id</name>
  <value>3998006</value>
</properties>
<properties>
  <name>end_time</name>
  <value>2014-01-10 14:48:00</value>
</properties>
<properties>
  <name>delivery_window_start</name>
  <value>13:30:00</value>
</properties>
```

```
<properties>
  <name>delivery_window_end</name>
  <value>14:30:00</value>
</properties>
<properties>
  <name>traveling_time</name>
  <value>30</value>
</properties>
<properties>
  <name>date</name>
  <value>2014-01-10</value>
</properties>
<properties>
  <name>resource_id</name>
  <value>44030</value>
</properties>
</activity>
</ns1:update_activity_response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.3 'get_file' Method

The 'get_file' method is used to retrieve file property details from ETAdirect.

4.3.1 'get_file' Request

The 'get_file' method request specifies:

- entity the file property is assigned to
- label of the property to be retrieved

The 'get_file' request contains the following elements:

Field	Required	Type	Description
user	Yes	struct	'user' structure
entity_id	Yes	string	ID of ETAdirect entity, to which the processed file property is assigned (ID for activity, external ID for resource)
property_id	Yes	string	label of file property for which data is to be retrieved

- **'get_file' Request Example**

The example is to retrieve property 'TECH_PHOTO' for entity 'TechX'.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:get_file>
      <user>
        <now>2014-01-10T14:00:45Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>02e4f7380d440cd9389669d9af600447</auth_string>
      </user>
      <entity_id>TechX</entity_id>
      <property_id>TECH_PHOTO</property_id>
    </ns1:get_file>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.3.2 'get_file' Response

The 'get_file' response contains the data on the method success or failure and 'file_name', 'file_data' and 'file_mime_type' of the file defined in the request.

The response contains the following elements:

Field	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error_description (if 'result_code' is other than '0')
file_name	string	name of the file
file_data	base64 binary	file data in base64 encoding
file_mime_type	string	mime type of the file

- **'get_file' Response Example**

Please note, that for the sake of convenience, the 'file_data' in the example is not an actual complete file.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:get_file_response>
      <result_code>0</result_code>
      <file_name>dawg.png</file_name>
      <file_data>iVBORw0KGgoAAAANSUUhEUgAAABAAAAAQAYAAAAf8/9hAAAAABGdBtUEAAK/INw
WK6QAAABl0RVh0U29mdHdhcmUAQWRvYmUgSW1hZ2VSZWFkeXhJZTAAAIpSURBVDjLddM9aFRBFibh9
8zM3WyybnYVf4KSQjBJJVZBixhRixSaShtBMKUoWomgnaCxsJdgIQSstE4nEhNREgyoZYhpkogkuMa4
/3fuHIu7gpLd00wz52POMzMydu/Dy958dMwYioomIIgqDa+VnWrzebNUejY/NV6nQ8nlR4ufXt0fzm2
WgxUgqBInAWdhemGbpCWNN9/XN27PPb1QbRdgjEhPqap2ZUv5+iOwvJnweT1mT5djZKjI6Ej/udz+wt
1OJzAKYgWyDjJWyFghmzFsbtCY2gsTJwv09/Vc7RTgAEQgsqAKaoWsm8wu/z7a8B7vA8cHD3Fr+ktFg
spO3a+vrdVfNEulJ/NT4zWngCBYY1oqSghKI465fvYwW+VAatPX07IZmF7YfrC0uDE8emPmiloFkHYi
BKxAxhmSRPlzVva2FGOU2Ad2ap4zg92MDBXJZczFmdflx05VEcAZMGIIClZASdesS2cU/dcm4sTBArN
zXTcNakiCb3/HLRsn4Fo2qyXh3WqDXzUlcgYnam3Dl4Hif82dbOiyiBGstSjg4majEpl8rpcNUQUjgk
ia0M5GVALBEBFUwflEv12b/Hig6SmAlidTzhcse6E7LixAchAtwNVxc1MnhprN/
+lh0txErxrPZVdFdrDEEZHT6LWpTbtq+HLSDDiOm2o1uqlyOT37bIhHdKaXoL6pqhq24Dzd96/tUYGw
PSBVv7atFglaFIu5KLuPxeX/xsp7aR6AAAAAE1FTkSuQmCC</file_data>
      <file_mime_type>image/png</file_mime_type>
    </ns1:get_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.4 'get_resource' Method

The 'get_resource' method is used to retrieve information of a specified existing resource.

4.4.1 'get_resource' Request

The 'get_resource' requests specify the resource whose data is to be retrieved.

The 'get_resource' requests contain the following elements:

Name	Required	Type	Description
user	Yes	struct	' user ' structure
id	Yes	string	external ID of the resource whose data is to be retrieved
date	No	string	YYYY-MM-DD format date of the resource's calendar for which the working time may be returned in the response If the 'date' element is sent in the request, the response will contain the 'time_from', 'time_to' and 'non_working_reason' fields. Otherwise, these fields are not returned

- **'get_resource' Request Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toatech:ResourceManagement:1.0">
  <SOAP-ENV:Body>
    <ns1:get_resource>
      <user>
        <now>2014-01-10T14:03:33Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>cdc963f4c701648458a1b65ec3ced71a</auth_string>
      </user>
      <id>test_get_resource-2</id>
      <date>2014-01-10</date>
    </ns1:get_resource>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.4.2 'get_resource' Response

The response to the 'get_resource' request contains data on the method success/failure and all properties of the processed resource.

The response of the 'get_resource' method contains the following elements:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
properties	array	array of resource ' property ' elements each containing one of the resource's properties
users	array	array of 'login' elements that contain identifiers of the users assigned to the

		resource
workskills	array	array of 'workskill' and 'workskill_group' elements each corresponding to one of work skills or work skill groups assigned to the resource. It is only present in the response if an optional 'date' parameter is set in the request (resource work skills depend on the selected date because of teamwork) and if the work skill/work skill group list for resource is not empty.

- **'users' Array of 'get_resource' Response**

The 'users' array consists of 'login' elements containing logins of the users assigned to the resource.

Name	Type	Description
login	string	login of the user assigned to the resource

- **'workskills' Array of 'get_resource' Response**

The 'workskills' array may contain the following elements:

Name	Type	Description
workskill	struct	any of the work skills defined in ETAdirect which the resource has
workskill_group	string	any of the work skill groups defined in ETAdirect which the resource has

- **'workskill' Element**

The 'workskill' elements consist of the following:

Name	Type	Description
label	string	label of the work skill defined in ETAdirect
ratio	string	level of qualification of the work skill defined with the 'label' for the resource. values range: 0 – 100

- **'get_resource' Response Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toatech:ResourceManagement:1.0">
  <SOAP-ENV:Body>
    <ns1:get_resource_response>
      <result_code>0</result_code>
      <properties>
        <property>
          <name>id</name>
          <value>test_get_resource-2</value>
        </property>
        <property>
          <name>status</name>
          <value>active</value>
        </property>
        <property>
          <name>parent_id</name>
          <value>test_get_resource.bucket</value>
        </property>
      </properties>
    </ns1:get_resource_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
</property>
<property>
  <name>type</name>
  <value>PR</value>
</property>
<property>
  <name>name</name>
  <value>test_get_resource-2</value>
</property>
<property>
  <name>language</name>
  <value>en</value>
</property>
<property>
  <name>time_zone_diff</name>
  <value>-300</value>
</property>
<property>
  <name>time_zone</name>
  <value>Eastern</value>
</property>
<property>
  <name>date_format</name>
  <value>mm/dd/yy</value>
</property>
<property>
  <name>time_format</name>
  <value>24-hour</value>
</property>
</properties>
<users>
  <login>user</login>
</users>
  <workskills>
    <workskill>
      <label>01</label>
      <ratio>100</ratio>
    </workskill>
    <workskill>
      <label>02</label>
      <ratio>100</ratio>
    </workskill>
    <workskill_group>Group - Number 0</workskill_group>
  </workskills>
</ns1:get_resource_response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5 Transaction Errors

For each request a response is returned. If an error occurs in the course of transaction processing, the corresponding error response is returned. This can be a SOAP fault (wrong or unknown request is sent) or an error response (a valid request contains invalid element(s)).

5.1 SOAP Faults

In case of errors standard SOAP Faults are returned. Soap Fault contains the following fields:

Soap Fault field	Possible values of this field	Description
faultcode	<ul style="list-style-type: none"> Client Server 	<p>This field is always returned.</p> <ul style="list-style-type: none"> Client – means that the problem is with the request – either request has incorrect format, or invalid authentication info is supplied etc. Server – means that the problem is on ETAdirect side.
faultstring	<ul style="list-style-type: none"> Authentication Failed Unknown location Bad request format etc 	<p>This field is always returned.</p> <p>It contains human-readable description of error</p>
faultactor	<ul style="list-style-type: none"> DISPATCHER API-PORTAL <absent> 	<p>This field is optional.</p> <p>This field is for diagnostic purposes and may be ignored by the Client Application.</p> <p>It signifies which part of ETAdirect system generated the Soap Fault.</p>
detail	free-format element	<p>This field is optional.</p> <p>This field contains ETAdirect specific subfields depending on the module emitting the SOAP Fault.</p> <p>This field is for diagnostic purposes and may be ignored by the Client Application.</p>

- **SOAP Fault Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Bad Request</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5.2 Error Responses

All responses contain a result code and description, when applicable (when the 'result_code' is other than '0').

Field	Description
result_code	result of the performed operation 'result_code' is returned in every response. For a successful transaction 'result_code' = 0 is returned. If transaction fails, the 'result_code' > 0.
error_msg	more specific description of the error 'error_msg' is returned only if 'result_code' is other than 0.

- **Error Response Example**

Example of an error response to an Activity Management API request ('get_file' method):

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:get_file_response>
      <result_code>18</result_code>
      <error_msg>Invalid activity_id=2332ee</error_msg>
    </ns1:get_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example of an error response to a Resource Management API request ('get_resource' method):

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toatech:ResourceManagement:1.0">
  <SOAP-ENV:Body>
    <ns1:get_resource_response>
      <result_code>24</result_code>
      <error_msg>Cannot get the resource id 'nonexistent'</error_msg>
    </ns1:get_resource_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5.3 Error Codes

5.3.1 Activity Management API Errors

Error codes related to the Activity Management API methods ('search_activities', 'update_activity' and 'get_file'):

Code	error_msg Example	Description
0		no error. Request has been successfully processed
3	Authentication failed	user authentication was unsuccessful
9	Parameter 'position_in_route' is equal to 'activity_id'	wrong activity position within a route
18	Search failed: 1679041228: search_appointments: 'gfh' is not a valid select_count value	wrong value of a parameter in the request
19	Activity not found: id=7996012	requested object is not found
20	Can't update activity: 9382904: update_appointment: Data has been changed	record to be updated changed by another user
100	Internal error	all other cases

5.3.2 Resource Management API Errors

Error codes related to the Resource Management API method ('get_resource'):

Code	error_msg Example	Description
0		no error. Request has been successfully processed
1	Internal error	the application server is not available
3	Key field is not visible to caller: login	the specified field is not visible to the user with such 'login' according to permissions
23	The resource external_id is not set	external ID of the resource is empty
24	Cannot get the resource id '3331904DG'	the resource with the specified external ID is not found
33	Authentication failed	user authentication was unsuccessful
43	Invalid value of date: ". (Must be 'YYYY-MM-DD')	wrong value of a parameter in the request
100	No permission	the user has no permission to perform the operation

6 Migration from 4.4 to 4.5

The SelfCare functionality in version 4.5 has not changed as compared to version 4.4 except the following:

6.1 'user' Authentication Structure

The 'user' authentication structure is unchanged, except the 'company' field now accepts the 'instance name' instead of the 'company' name.