Sinch Contact Pro

Task Management Interface (TMI)

December 2022

Date	Description
22.5.2006	The original release version.
25.1.2007	Added UpdateTask, UpdateTaskProfiles.
5.2.2007	Updated UpdateTaskProfiles to support taskNames element.
9.2.2007	Added {Tree} InfoCard visible Question name (separated by | character).
26.6.2007	Added: Windows authentication, IIS 6 and ASP.NET 2.0
28.4.2011	Added: Priority special handling
16.8.2011	Updated from the version 6.0 to the version 7.0.
6.12.2012	Added script result functions for the version 7.0 SP03.
27.5.2014	Changed BCM to SAP Contact Center
22.1.2015	Added detailed information about ECF task with type XRI
18.3.2016	Added Required Agent handling
27.5.2016	Added <queueName> to TaskItem.
19.8.2016	Removed Test.asmx, fixed registry settings, added <queueName> to CreateTask parameters.
29.3.2017	Added <datatype> to ScriptResultDetail.
10.11.2017	Improved <priority> description.
4.12.2017	Added 0 => null conversion to <priority> description.
4.1.2018	Document template changed.
26.9.2018	Added information about sub type for ECF tasks.
15.12.2022	Changed Contact Center to Sinch Contact Pro

Contents

1 Introduction

This document describes *Task Management Interface* (TMI) of the Sinch Contact Pro software. This information is directed to systems integrators and third-party software vendors who wish to create new tasks to the Sinch Contact Pro system and read task information from the system.
Details of the interface and this document may be changed without prior notice, but the basic principles of the interface (such as use of SOAP over HTTP) are not subject to change.

1.1 Security

The interface provides a powerful tool to browse and manipulate data in the Sinch Contact Pro system, and therefore you should protect the interface carefully from malicious use.
In Integration Interfaces installation, TMI is disabled by default. There are also several options for TMI authentication (basic, certification, anonymous). If a Sinch Contact Pro user is authenticated, then queue right MANAGE_CONTACTS (Manage Contact History) is checked when creating new tasks, updating tasks or retrieving tasks. Script result access rights depend also on queue rights, so no Script access rights are used.

1.2 SOAP

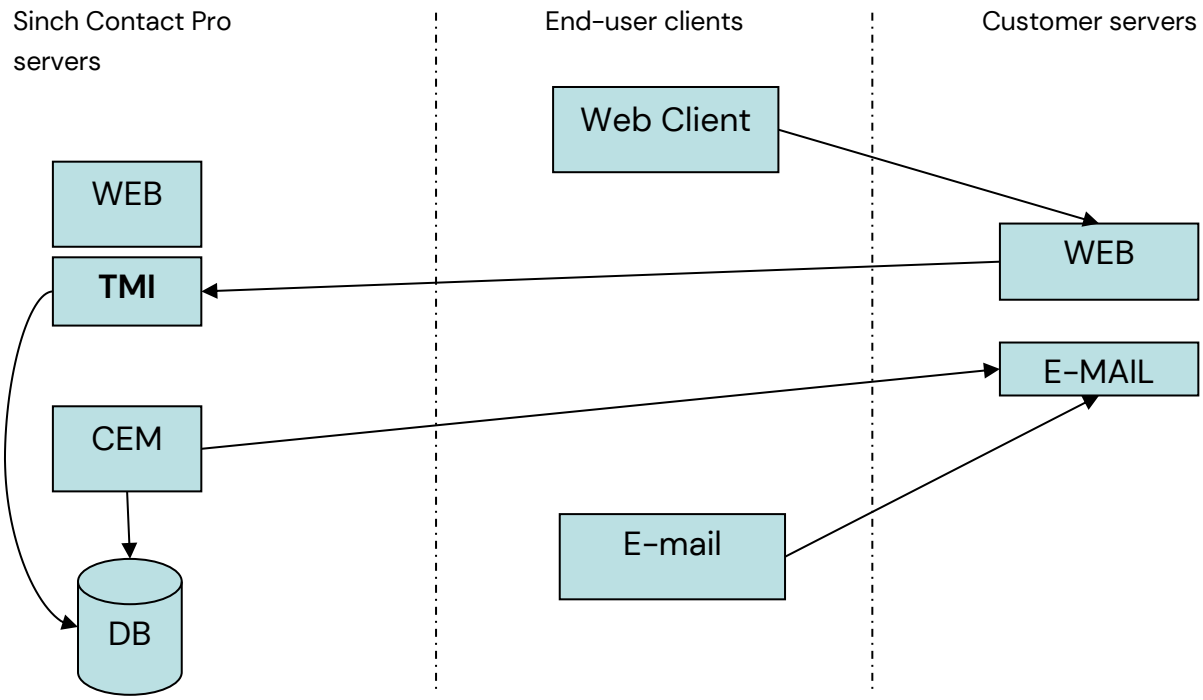
The interface is based on the SOAP version 1.1 (http://www.w3.org/TR/SOAP). The interface uses the HTTP protocol to carry SOAP messages between a SOAP client and itself.
If an operation fails for some reason, a standard SOAP 1.1 fault message is returned. In the current version of the interface only the faultcode and faultstring elements are used. The possible fault codes are only Client or Server without any further classification of the error. However, the faultstring element is used for returning a comprehensible description of the error situation.

```
HTTP/1.0 200 OK
Server: WWS 0.1
Date: Wed, 17 Sep 2003 21:30:59 GMT
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-type: text/xml
Content-Length: 238

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Body>
      <env:Fault>
         <faultcode>env:Client</faultcode>
         <faultstring>Unknown reason:whatever</faultstring>
      </env:Fault>
   </env:Body>
</env:Envelope>
```

2 Architecture Overview

The interface is a server-side component running under a web server (IIS). It connects to the Sinch Contact Pro Operative database by using ADO.NET.
The following graphics shows typical use of the interface. In this case the customer has its own web server that creates new tasks into the Operative database in the same way as if they had arrived from an e-mail channel.



TMI can also be used for retrieving task information from the Operative database by using varying criteria (new or modified tasks, task by ID (GUID), etc.).
Tasks can also be updated via TMI.

TMI is updated in Contact Center 7.0 SP03 to support script results. Script results are similar to the version 6.0 info cards. The version 7.0 does not have info cards or activity profiles, so script results can be used instead to have a predefined set of question – answer pairs.

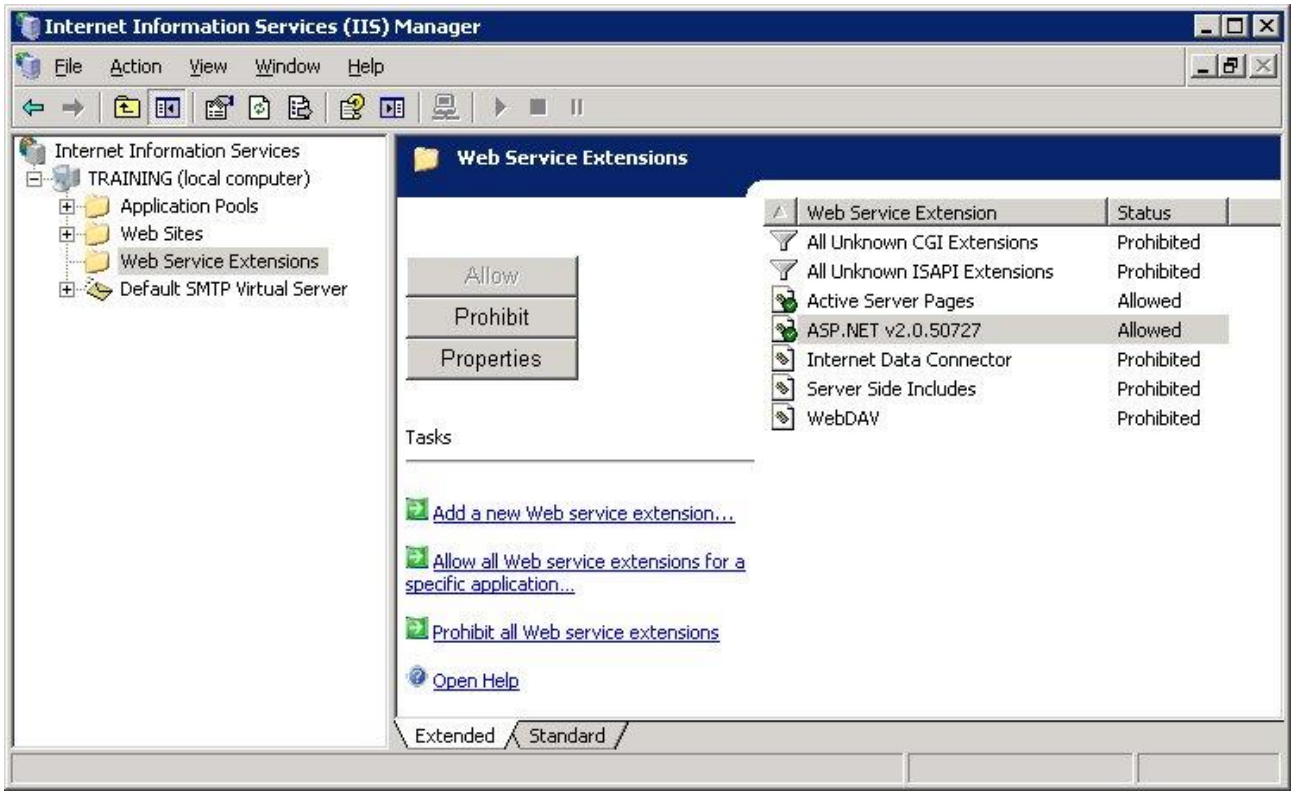
The version 6.0 TMI had one function to set info card definitions. The version 7.0 does not have a function to set Script definitions, but instead SC Import/Export can be used.

ECF tasks can be created in version 7.0 SP08. Task type must be XRI when ECF is used.

CCtr 1705 adds <dataType> field to ScriptResultDetail.

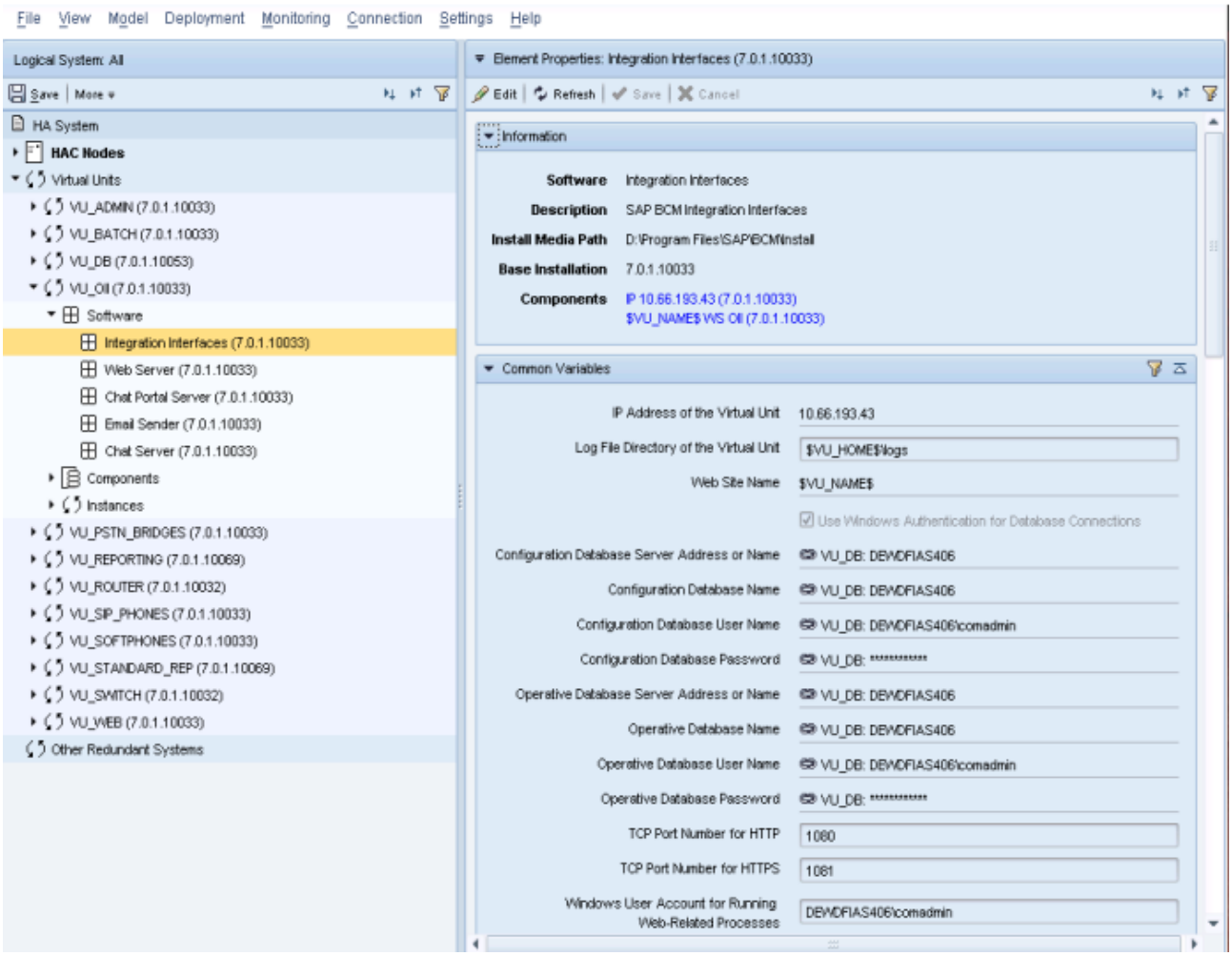
3 Installation

To install the interface, install with the IA tool the Integration Interfaces package to a virtual unit where also the Sinch Contact Pro Web Server is installed. The server needs to have IIS 5.1 (or later) and .NET Framework 4.5.2 (or later). ASP.NET v4.0.30319 must be allowed in IIS.



3.1 IA Setup

In the Sinch Contact Pro Infrastructure Administrator (IA) tool, configure the TMI-specific variables. TMI uses Configuration and Operative database settings.
TMI-specific variables are saved to registry under TMI. For more information, see the chapter *Configuration*.

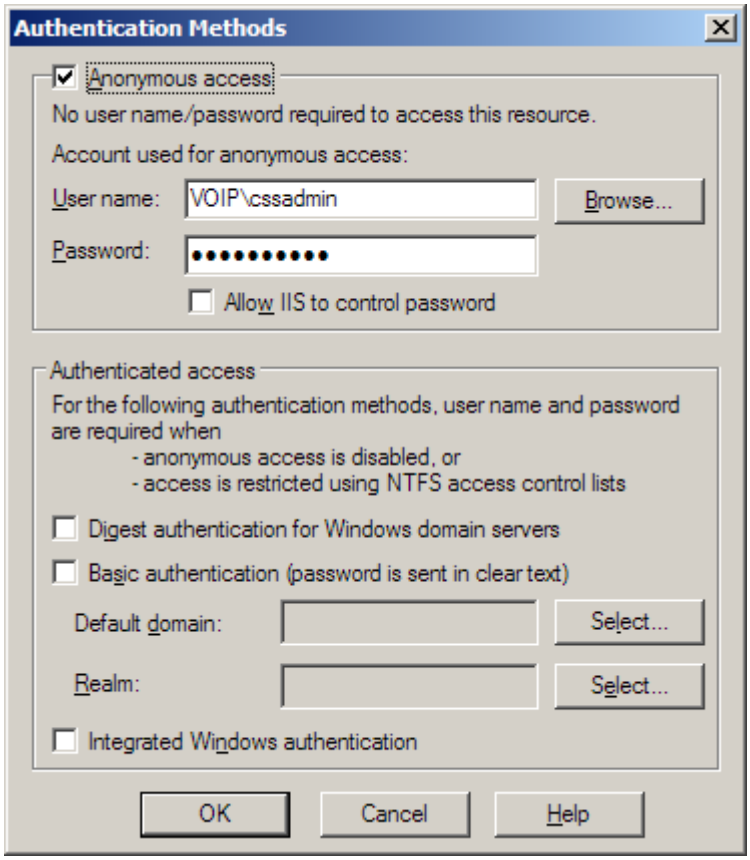


3.2 Windows Authentication

TMI uses Windows authentication to access Configuration and Operative database (using VU registry settings `CONFIGURATION_DSN` and `OPERATIVE_DSN`), if UseWindowsAuthentication VU registry setting is "yes" or if SQL Server UID is not given. In order for this to work, Web.config must contain

```
<identity impersonate="true"/>
```

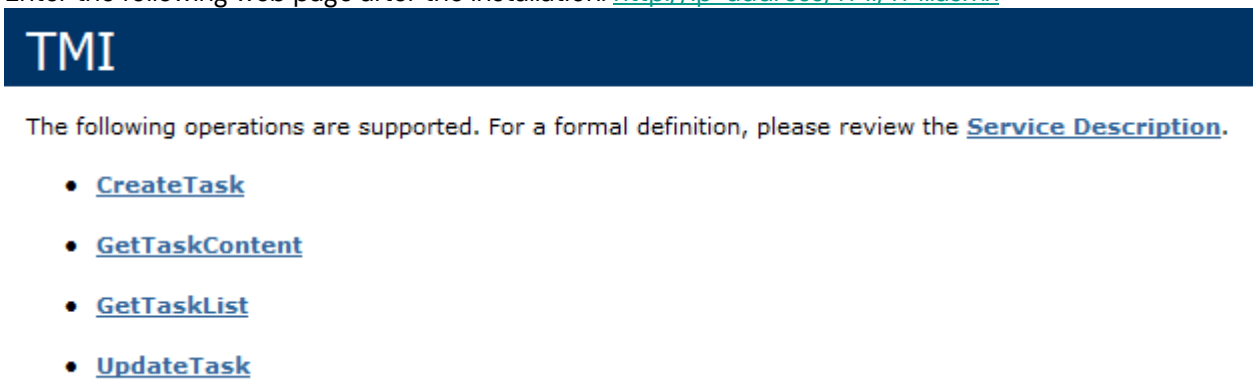
under <system.web>. This is done automatically by setup.
The UID used in Windows authentication is defined in IIS / TMI / Properties / Directory Security / Anonymous access and authentication control / Edit / Anonymous access:



3.3 Testing Installation

3.3.1 TMI.asmx

Enter the following web page after the installation: http://ip-address/TMI/TMI.asmx

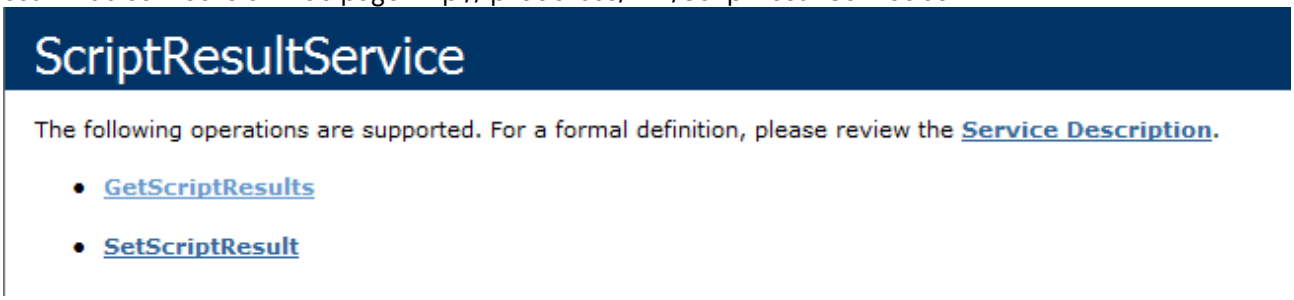


TMI page contains the following functions:

CreateTask	Creates a new task.
GetTaskContent	Retrieves the contents of a task.
GetTaskList	Retrieves a list of tasks.
UpdateTask	Updates task contents.

3.3.2 ScriptResultService.asmx

Script result web service is on web page http://ip-address/TMI/ScriptResultService.asmx

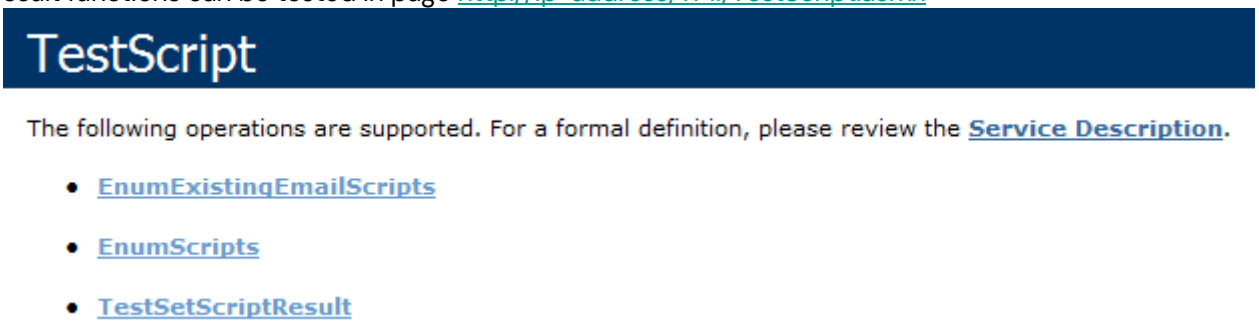


ScriptResultService.asmx page contains the following functions:

GetScriptResults	Retrieves script results from a task.
SetScriptResult	Sets script result for a task.

3.3.3 TestScript.asmx

Script result functions can be tested in page http://ip-address/TMI/TestScript.asmx



TestScript.asmx page contains the following functions:

EnumExistingEmailScripts	Displays a listing of existing recent e-mails and their scripts. The purpose of this testing function is to get contactId (e-mail/task id) and scriptName that can be used in other test functions. It also displays the html code that is saved to script result, and which is displayed in CDT history view.
EnumScripts	Displays all scripts, script questions and script answers that in the current system.
TestSetScriptResult	Can be used to test SetScriptResult function, which has a complex data type, so it cannot be tested directly, unlike GetScriptResults function.

3.4 Stopping TMI

The safe way to stop TMI is to modify the Web.config file among the virtual unit's program files (for example, in C:\SAP\BCM\VU\VU1\web\TMI).

4 Configuration

4.1 Virtual Unit Registry Settings

The following variables are set in registry on VU level during installation of the Integration Interfaces package with the IA tool:

Setting	Default value	Description
CONFIGURATION_DSN		Data source name for Configuration database.
OPERATIVE_DSN		Data source name for Operative database.
IP		IP address of virtual unit.
WebPort	80	Port of web server.

4.2 TMI-Specific Registry Settings

The following registry settings are added during installation by the IA tool into the key HKEY_LOCAL_MACHINE\SOFTWARE\Wicom\VU1\TMI.

Setting	Default value	Description
InterfaceEnabled	0	IA variable: TMI_INTERFACE_ENABLED By default TMI is not enabled.
BasicAuthenticationEnabled	0	IA variable: TMI_BASIC_AUTHENTICATION_ENABLED Enables using HTTP basic authentication scheme for incoming TMI requests.
CertificateAuthenticationEnabled	0	IA variable: TMI_CERTIFICATE_AUTHENTICATION_ENABLED Authenticates incoming TMI requests with client certificate.
AnonymousAuthenticationMode	disabled	IA variable: TMI_ANONYMOUS_AUTHENTICATION_MODE Defines whether incoming TMI requests may proceed without authentication. disabled = Do Not Allow local = Allow Local Anonymous Requests

		remote = Allow Local and Remote Anonymous Requests noauthentication = Allow All Requests without Authentication
AnonymousAuthenticationLoginId		IA variable: TMI_ANONYMOUS_AUTHENTICATION_LOGINID Logon ID for Anonymous TMI Requests. Use only when local or remote anonymous requests are allowed.
LogLevel	warning	Log level.

5 TMI Interface Specification

TMI interface contains the following operations:

- **CreateTask** => Creates a new task.
- **GetTaskContent** => Retrieves the contents of a task.
- **GetTaskList** => Retrieves a list of tasks.
- **UpdateTask** => Update task contents.

Each operation consists of two messages: the request and the response.
If a Sinch Contact Pro user is authenticated, then queue right MANAGE_CONTACTS (Manage Contact History) is checked when creating new tasks, updating tasks or retrieving tasks.

5.1 CreateTask Operation

5.1.1 Definition

Purpose	The operation creates a new task.
Request name	CreateTask
Request elements	TaskData <data> => Contains several sub-elements which can be null or empty. Skill array <skills> => Array of skills and skill values. RequiredAgents <requiredAgents> => Contains expiry time and array of required agents. Either GUID for user or user login ID can be used to identify the agent. The following TaskData sub-elements are the most important: String <item.id> => GUID. Can be empty or can contain an external ID from the SOAP client. This will identify the new task. String <item.type> => Default value is “TASK“. String <item.queue> => GUID for e-mail queue. Can also be queue name (case sensitive) or queue address in input. String <item.queueName> => Queue name (case sensitive). Use either <queue> or <queueName>. String <item.responsible> => GUID for user. Can also contain user login ID in input. Can be empty, in which case the task is allocated by CEM (from queue). String <item.status> => Default value is “New”. Other possible values: "Allocated", "Forwarded", "Pending", "Open", "Handled", "Deleted".

	String <subject> => Subject of the task. String <solution> => Body text of an e-mail message. Attachment array <attachments> => Can contain the task attachments.
Response name	CreateTaskResponse
Response elements	TaskItem <response> => Contains the following sub-elements: String <id> => GUID. This identifies the new task. String <type> => Default value is “EMAIL“ String <queue> => GUID for queue. String <queueName> => Queue name. String <responsible> => GUID for user. String <status> => Default value is “New”. Other possible values: "Allocated", "Forwarded", "Pending", "Open", "Handled", "Deleted". String <rowVersion> => Hex string containing the database version information for the created task.

5.1.2 CreateTask Message

This message carries a CreateTask operation request and its parameters from a SOAP client to the TMI interface.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateTask xmlns="urn:TMI">
      <data>
        <item>
          <queue>EmailQueue</queue>
        </item>
        <subject>Card does not work</subject>
        <body>Card number: 4920 1234 5678 2345
          Phone number: 123456
          E-mail address: asdf@gmail.com
          Feedback: Card does not work. New card required.
        </body>
      </data>
      <skills>
        <Skill>
          <id>CreditCard</id>
          <value>4</value>
        </Skill>
      </skills>
      <requiredAgents>
        <expiryTime>3600</expiryTime>
        <agents>
          <RequiredAgent>
            <login>user1</login>
          </RequiredAgent>
        </agents>
```

```
      </requiredAgents>
    </CreateTask>
  </soap:Body>
</soap:Envelope>
```

5.1.3 CreateTaskResponse Message

This message carries results of a CreateTask operation from the TMI interface back to a SOAP client.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateTaskResponse xmlns="urn:TMI">
      <response>
        <id>779019BC8FA4483FBCDF0A88C611BB9F</id>
        <type>EMAIL</type>
        <queue>001EAA7D6A1E4E4889FC03F17F649E3C</queue>
        <queueName>EmailQueue</queueName>
        <responsible></responsible>
        <status>New</status>
        <rowVersion>00000000027E78D2</rowVersion>
      </response>
    </CreateTaskResponse>
  </soap:Body>
</soap:Envelope>
```

5.2 GetTaskContent Operation

5.2.1 Definition

Purpose	The operation retrieves the content of a given task by task id. It is possible to filter the sub-elements to be returned.
Request name	GetTaskContent
Request elements	String <id> => GUID. The id of the task. String <filter> => A character array which can contain the values 1 (the element should be returned) and 0 (the element should not be returned). The elements are the following: Item and other top-level sub-elements (“1”) Key-value pairs (“01”) Attachment names (“001”) Attachment content (“0001”) All sub-elements (empty or “1111”)
Response name	GetTaskContentResponse
Response elements	TaskData <response> => Contains all the requested data for the task.

5.2.2 GetTaskContent Message

This message carries a GetTaskContent operation request and its parameters from a SOAP client to the TMI interface.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTaskContent xmlns="urn:TMI">
      <id>779019BC8FA4483FBCDF0A88C611BB9F</id>
      <filter/>
    </GetTaskContent>
  </soap:Body>
</soap:Envelope>
```

5.2.3 GetTaskContentResponse Message

This message carries results of a GetTaskContent operation from the TMI interface back to a SOAP client.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTaskContentResponse xmlns="urn:TMI">
      <response>
        <item>
          <id>779019BC8FA4483FBCDF0A88C611BB9F</id>
          <type>EMAIL</type>
          <queue>001EAA7D6A1E4E4889FC03F17F649E3C</queue>
          <queueName>EmailQueue</queueName>
          <status>Open</status>
          <rowVersion>00000000027E78D5</rowVersion>
        </item>
        <caseId />
        <cid>CID45409737AE07</cid>
        <created>2011-08-16T12:54:19.113</created>
        <priority xsi:nil="true" />
        <warningTime xsi:nil="true" />
        <criticalTime>2011-08-23T12:54:19.12</criticalTime>
        <fromAddress />
        <toAddress />
        <firstModified>2011-08-16T12:54:19.113</firstModified>
        <lastModified>2011-08-16T12:54:19.113</lastModified>
        <solutionDate xsi:nil="true" />
        <subject>Card does not work</subject>
        <body>
          Card number: 4920 1234 5678 2345
          Phone number: 123456
          E-mail address: asdf@gmail.com
          Feedback: Card does not work. New card required.
        </body>
        <customer />
        <index>519256</index>
        <values>
          <KV>
            <k>CEM_DATA</k>
            <v>CEM_Received=2011-08-16+12%3a54%3a19;CEM_GUID=*%7b779019BC-8FA4-483F-
BCDF-0A88C611BB9F%7d;</v>
          </KV>
```

```
        </values>
      </response>
    </GetTaskContentResponse>
  </soap:Body>
</soap:Envelope>
```

5.3 GetTaskList Operation

5.3.1 Definition

Purpose	The operation retrieves the task items for the required tasks. It is possible to give complex filters in the TaskItem element. This operation can also be used for polling modified tasks.
Request name	GetTaskList
Request elements	Int32 <top> => The maximum number of tasks to be retrieved. If null, there is no limit. String <modified> => The hexadecimal string containing database rowVersion value. Returns only the tasks modified at this point of time and after it (which are dated before the current date in the database server). Use this element to poll the modified tasks (for example, once in every ten seconds). Give the returned serverDate as a parameter the next time. TaskItem <item> => Each sub-element can be used as a filter. By default the element must match exactly (i.e. case-insensitive comparison), but it is also possible to have patterns, between values, and value lists by using ~ as the first character. If ~ is followed by the ! character, the match is negated: ~!=Open means NOT equal to "Open", ~!_ means NOT BETWEEN, etc. ~=value => The "value" must match exactly (case-insensitive). Use this if the the value itself contains ~ as the first character. ~~value => The "value" is a pattern, and the LIKE database operator is used. ~-value1|value2 => Matches if the value >= value1 and the value < value2. ~_value1|value2 => Matches if the value >= value1 and the value <= value2 (BETWEEN). ~.value1|value2|value3 => Matches if the value is one of the given values (case-insensitive). In practice, only item.type and item.status can have pattern criteria, because other elements (id, queue, responsible) are GUIDs. item.queue accepts queue GUID, name or e-mail address (which are converted to single GUID for search). item.queueName accepts queue name (which is converted to single GUID for search).
Response name	GetTaskListResponse
Response elements	String <serverDate> => The database server date in hexadecimal string format (for example, 00000000027E78D5). This should be given in a <modified> parameter next time if the client wants to poll modified tasks. TaskItem array <items> => The list of matching tasks in miscellaneous order.

5.3.2 GetTaskList Message

This message carries a GetTaskList operation request and its parameters from a SOAP client to the TMI interface.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTaskList xmlns="urn:TMI">
      <top>2</top>
      <modified>0000000000484C3E</modified>
      <item>
        <type>EMAIL</type>
        <queue>03FF0CDC47B649508C0FB511AF41333F</queue>
        <responsible>F4887E499FEE453F9600863EAAA996C7</responsible>
        <status>Handled</status>
      </item>
    </GetTaskList>
  </soap:Body>
</soap:Envelope>
```

5.3.3 GetTaskListResponse Message

This message carries results of a GetTaskList operation from the TMI interface back to a SOAP client.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTaskListResponse xmlns="urn:TMI">
      <response>
        <serverDate>00000000027E90C9</serverDate>
        <items>
          <TaskItem>
            <id>C4ADBBB108F7DF11B65A0003FF38AC85</id>
            <type>EMAIL</type>
            <queue>03FF0CDC47B649508C0FB511AF41333F</queue>
            <queueName>EmailQueue</queueName>
            <responsible>F4887E499FEE453F9600863EAAA996C7</responsible>
            <status>Handled</status>
            <rowVersion>0000000000484C3E</rowVersion>
          </TaskItem>
          <TaskItem>
            <id>05020FF50CF7DF11B65A0003FF38AC85</id>
            <type>EMAIL</type>
            <queue>03FF0CDC47B649508C0FB511AF41333F</queue>
            <queueName>EmailQueue</queueName>
            <responsible>F4887E499FEE453F9600863EAAA996C7</responsible>
            <status>Handled</status>
            <rowVersion>000000000048526A</rowVersion>
          </TaskItem>
        </items>
      </response>
    </GetTaskListResponse>
  </soap:Body>
</soap:Envelope>
```

5.4 UpdateTask Operation

5.4.1 Definition

Purpose	The operation updates task data.
Request name	UpdateTask
Request elements	TaskData <data> => Contains several sub-elements which can be null or empty. Usually the information is got from GetTaskContent operation. The following sub-elements are the most important: String <item.id> => GUID. Used to identify the task. String <item.type> => Ignored in UpdateTask, value can't be changed. String <item.queue> => GUID for e-mail queue. Can also contain queue name (case sensitive) or queue e-mail address. String <item.queueName> => Ignored in UpdateTask operation. String <item.responsible> => GUID for user. String <item.status> => "New", "Allocated", "Forwarded", "Pending", "Open", "Handled", "Deleted". String <subject> => Subject of the task. String <body> => Body text of an e-mail message. Attachment array <attachments> => Attachments.
Response name	UpdateTaskResponse
Response elements	TaskItem <item> => Contains the new rowVersion and other information.

5.4.2 UpdateTask Message

This message carries an UpdateTask operation request and its parameters from a SOAP client to the TMI interface.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateTask xmlns="urn:TMI">
      <data>
        <item>
          <id>string</id>
          <type>string</type>
          <queue>string</queue>
          <responsible>string</responsible>
          <status>string</status>
          <rowVersion>string</rowVersion>
        </item>
```

```
        <caseId>string</caseId>
        <cid>string</cid>
        <subject>string</subject>
        <created>dateTime</created>
        <creator>string</creator>
        <priority>int</priority>
        <warningTime>dateTime</warningTime>
        <criticalTime>dateTime</criticalTime>
        <fromAddress>string</fromAddress>
        <toAddress>string</toAddress>
        <toAddressAnswer>string</toAddressAnswer>
        <ccAddressAnswer>string</ccAddressAnswer>
        <bccAddressAnswer>string</bccAddressAnswer>
        <firstModifier>string</firstModifier>
        <firstModified>dateTime</firstModified>
        <lastModifier>string</lastModifier>
        <lastModified>dateTime</lastModified>
        <solutionDate>dateTime</solutionDate>
        <body>string</body>
        <customer>string</customer>
        <index>long</index>
        <values>
          <KV>
            <k>string</k>
            <v>string</v>
          </KV>
          <KV>
            <k>string</k>
            <v>string</v>
          </KV>
        </values>
        <attachments>
          <Attachment>
            <id>string</id>
            <filename>string</filename>
            <creator>string</creator>
            <created>string</created>
            <ext>string</ext>
            <size>long</size>
            <data>base64Binary</data>
          </Attachment>
          <Attachment>
            <id>string</id>
            <filename>string</filename>
            <creator>string</creator>
            <created>string</created>
            <ext>string</ext>
            <size>long</size>
            <data>base64Binary</data>
          </Attachment>
        </attachments>
      </data>
    </UpdateTask>
  </soap:Body>
</soap:Envelope>
```

5.4.3 UpdateTaskResponse Message

This message carries results of an UpdateTask operation from the TMI interface back to a SOAP client.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateTaskResponse xmlns="urn:TMI">
      <response>
        <id>string</id>
        <type>string</type>
        <queue>string</queue>
        <queueName>string</queueName>
        <responsible>string</responsible>
        <status>string</status>
        <rowVersion>string</rowVersion>
      </response>
    </UpdateTaskResponse>
  </soap:Body>
</soap:Envelope>
```

6 TMI Content Description

This section describes the contents of the XML elements that carry task information from a Sinch Contact Pro system via the interface.

6.1 GUID Formatting in Input and Output

IDs like TaskItem.id, TaskItem.queue and TaskItem.responsible are GUIDs internally, but Strings in SOAP interface. GUID format is UPPERCASE and without - characters, so exactly 32 characters, like `C4ADBBB108F7DF11B65A0003FF38AC85` in output always. In input to TMI interface, the hexadecimal characters can be in lowercase also, and there can be - characters, so the following format is accepted as well: `c4adbbb1-08f7-df11-b65a-0003ff38ac85`
Additionally, some elements accept name instead of GUID in input:

- TaskItem.queue (queue name or queue e-mail address)
- TaskItem.responsible (user login name/id)
- Skill.id (skill external name)
- RequiredAgents.requiredAgent.login (user login name)

6.2 TaskItem Element

6.2.1 Definition

Purpose	The element represents the basic information for a task.
Element name	TaskItem
Scope	CreateTask, GetTaskContent, GetTaskList, UpdateTask
Sub-elements	String <id> => GUID, identifies the task. String <type> => The name of the task, by default "TASK". Other values: "EMAIL" (created by CEM from e-mail queue), "ACTION"(action items used in CRM integration in OII), "EMAILOUT" (sent e-mail), "XRI" (used in ECF). String <queue> => GUID of the queue. String <queueName> => Queue name. String <responsible> => GUID of the user. String <status> => One of the following values: New => The default value for a new task.

	Allocated => CEM has allocated task for some user. Forwarded => Task has been forwarded to some other user. Pending => Task is allocated to user but user is not currently processing it. Does not prohibit CEM allocating new tasks to user. Open => Task is being processed by some user. Handled => Task has been handled. Deleted => Task has been deleted (is automatically purged from the database later). String <rowVersion> => Hexadecimal database RowVersion indicating when the task was modified last time.

6.2.2 XML Schema Description

```
<s:complexType name="TaskItem">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="type" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="queue" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="queueName" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="responsible" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="status" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="rowVersion" type="s:string" />
  </s:sequence>
</s:complexType>
```

6.3 TaskData Element

6.3.1 Definition

Purpose	The element represents a single task (the whole task content).
Element name	TaskData
Scope	CreateTask, GetTaskContent, UpdateTask
Sub-elements	TaskItem <item> => Contains the basic information which can be used to identify (id) and monitor task modifications (rowVersion). Also contains the current queue and responsible user. String <caseId> => Case ID for the task, integer. CEM saves the Index of the original task to this field, if this task is response to some other task. String <cid> => Another Case ID for the task in format: CIDDB8659A23124. CEM generates this for incoming e-mails.

	DateTime <created> => DateTime value indicating when the task was created into the database. String <creator> => GUID, user who created the task. Int32 <priority> => Priority value for the task. Positive integer value that affects how CEM allocates the task. Integer value is a divider for the waiting time. The task having the lower value is allocated first (but it also depends on task creation time). Null value means that queue priority is used instead (with a default value of 1). It is not possible to give null value for priority when creating the task, and that's why TMI changes value 0 to null (which then CEM changes to queue priority). Contact allocation priority = [skill value] * [Importance of Skill Matching] + ([contact waiting time]/[queue priority])* [Importance of Contact Waiting Time] DateTime <warningTime> => DateTime value which is set by CEM when task is created. In CDT the task goes yellow when this time is reached. Value is taken from queue settings, no default. DateTime <criticalTime> => DateTime value which is set by CEM when task is created. In CDT the task goes red when this time is reached. Value is taken from queue settings, and the default value is 1 week after <created> time. String <fromAddress> => E-mail address of the sender. String <toAddress> => E-mail address of the queue. String <toAddressAnswer> => TO e-mail address when answering to the e-mail. String <ccAddressAnswer> => CC e-mail address when answering to the e-mail. String <bccAddressAnswer> => BCC e-mail address when answering to the e-mail. String <firstModifier> => GUID, user who first modified the task. DateTime <firstModified> => DateTime when the task was first modified. String <lastModifier> => GUID, user who last modified the task. DateTime <lastModified> => DateTime when the task was last modified. DateTime <solutionDate> => DateTime when the solution was given. String <subject> => Subject of an e-mail. String <body> => Body text of an e-mail. String <customer> => Customer name. Int64 <index> => Automatically generated integer ID for task. KV array <values> => An array of KV (key-value) elements. Attachment array <attachments> => An array of attachment elements.

6.3.2 XML Schema Description

```
<s:complexType name="TaskData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="item" type="s0:TaskItem" />
    <s:element minOccurs="0" maxOccurs="1" name="caseId" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="cid" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="subject" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="created" nillable="true"
type="s:dateTime" />
    <s:element minOccurs="0" maxOccurs="1" name="creator" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="priority" nillable="true"
type="s:int" />
    <s:element minOccurs="1" maxOccurs="1" name="warningTime" nillable="true"
type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="criticalTime" nillable="true"
type="s:dateTime" />
    <s:element minOccurs="0" maxOccurs="1" name="fromAddress" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="toAddress" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="toAddressAnswer" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="ccAddressAnswer" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="bccAddressAnswer" type="s:string"
/>
    <s:element minOccurs="0" maxOccurs="1" name="firstModifier" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="firstModified" nillable="true"
type="s:dateTime" />
    <s:element minOccurs="0" maxOccurs="1" name="lastModifier" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="lastModified" nillable="true"
type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="solutionDate" nillable="true"
type="s:dateTime" />
    <s:element minOccurs="0" maxOccurs="1" name="body" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="customer" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="index" nillable="true"
type="s:long" />
    <s:element minOccurs="0" maxOccurs="1" name="values" type="s0:ArrayOfKV" />
    <s:element minOccurs="0" maxOccurs="1" name="attachments"
type="s0:ArrayOfAttachment" />
  </s:sequence>
</s:complexType>
```

6.4 KV Element

6.4.1 Definition

Purpose	The element represents one key-value pair.
Element name	KV
Scope	CreateTask, GetTaskContent, UpdateTask
Sub-elements	String <k> => The key. Key "CEM_DATA" is reserved for CEM internal use. String <v> => The value.

6.4.2 XML Schema Description

```
<s:complexType name="KV">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="k" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="v" type="s:string" />
  </s:sequence>
</s:complexType>
```

6.4.3 Usage

The following table lists the commonly used key-value pairs. Some of them will automatically get a value when a new task is created.

Key	Value
From	The e-mail address of the sender.
FromName	The name of the sender.
To	E-mail address of receiver (queue).
CC	CC address(es).
BCC	BCC address(es).
ReplyAddress	E-mail address where the reply should be sent.

6.5 Attachment Element

6.5.1 Definition

Purpose	The element represents one file attached to the task.
Element name	Attachment
Scope	CreateTask, GetTaskContent, UpdateTask
Sub-elements	String <id> => GUID for the attachment. String <filename> => The original file name (without the path), including the extension: 5589B21EB806E011835800026554D1252.htm String <creator> => Creator of the attachment. DateTime <created> => DateTime when attachment was created.

	String <ext> => Extension of the file name (without . character): htm Int64 <size> => Size of the attachment in bytes. Byte array <data> => Attachment data.
Notes	When creating a new attachment, only give <filename> and <data>, the rest is generated automatically. To delete existing attachment, give null as <data>.

6.5.2 XML Schema Description

```
<s:complexType name="Attachment">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="filename" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="creator" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="created" nillable="true"
type="s:dateTime" />
    <s:element minOccurs="0" maxOccurs="1" name="ext" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="size" type="s:long" />
    <s:element minOccurs="0" maxOccurs="1" name="data" type="s:base64Binary" />
  </s:sequence>
</s:complexType>
```

6.6 Skill Element

6.6.1 Definition

Purpose	The element represents one skill-value pair.
Element name	Skill
Scope	CreateTask
Sub-elements	String <id> => GUID or external name of the skill. Int32 <value> => The skill value (1-5).

6.6.2 XML Schema Description

```
<s:complexType name="Skill">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="value" type="s:int" />
  </s:sequence>
</s:complexType>
```

6.7 RequiredAgents Element

6.7.1 Definition

Purpose	The element represents required agents.
Element name	RequiredAgents
Scope	CreateTask
Sub-elements	Int32 <expiryTime> => Expiry time for the required agents in seconds. RequiredAgent array <agents> => An array of required agent elements.

6.7.2 XML Schema Description

```
<s:complexType name="RequiredAgents">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="expiryTime" nillable="true"
type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="agents"
type="s0:ArrayOfRequiredAgent" />
  </s:sequence>
</s:complexType>
```

6.8 RequiredAgent Element

6.8.1 Definition

Purpose	The element represents one required agent.
Element name	RequiredAgent
Scope	CreateTask
Sub-elements	Either GUID for user or user login ID can be used to identify the agent. String <id> => GUID for user. String <login> => User login ID.

6.8.2 XML Schema Description

```
<s:complexType name="RequiredAgent">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="login" type="s:string" />
  </s:sequence>
</s:complexType>
```

7 ECF Tasks

In CCtr 7 SP08, support for ECF tasks is added to TMI. Task needs to have the following information when created with CreateTask operation:

- Task type must be “XRI”
- Task needs to have a queue (of type e-mail queue)
- The e-mail address of sender (fromAddress)
- The e-mail address of queue (toAddress)
- Subject

There can be two optional key-value pairs:

- K = “ChannelSubType”, V = “Ticket (for example)”.
  NOTE: Do not use sub type action for ECF tasks. That is meant for action items received from CRM WebClient.
- K = “From”, V = “e-mail address of sender for reporting, usually the same as fromAddress”

Here is a sample CreateTask request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateTask xmlns="urn:TMI">
      <data>
        <item>
          <type>XRI</type>
          <queue>E-Mail Queue 1</queue>
        </item>
        <fromAddress>customer@work.corp</fromAddress>
        <toAddress>email.queue1@cctr.sap</toAddress>
        <subject>Extremely Complicated Fun</subject>
        <body>Can be used for e-mail keyword routing by CEM</body>
        <values>
          <KV>
            <k>From</k>
            <v>customer@work.corp</v>
          </KV>
          <KV>
            <k>ChannelSubType</k>
            <v>Ticket</v>
          </KV>
        </values>
      </data>
    </CreateTask>
  </soap:Body>
</soap:Envelope>
```

8 TMI WSDL Description

The WSDL (http://www.w3.org/TR/wsdl) description of the interface is the following:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s0="urn:TMI"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://sap.com/bcm/TMI"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://sap.com/bcm/TMI"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="urn:TMI">
      <s:element name="CreateTask">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="data" type="s0:TaskData" />
            <s:element minOccurs="0" maxOccurs="1" name="skills"
type="s0:ArrayOfSkill" />
            <s:element minOccurs="0" maxOccurs="1" name="requiredAgents"
type="s0:RequiredAgents" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="TaskData">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="item" type="s0:TaskItem" />
          <s:element minOccurs="0" maxOccurs="1" name="caseId" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="cid" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="subject" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="created" nillable="true"
type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="creator" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="priority" nillable="true"
type="s:int" />
          <s:element minOccurs="1" maxOccurs="1" name="warningTime" nillable="true"
type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1" name="criticalTime" nillable="true"
type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="fromAddress" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="toAddress" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="toAddressAnswer"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="ccAddressAnswer"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="bccAddressAnswer"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="firstModifier"
type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="firstModified"
nillable="true" type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="lastModifier" type="s:string"
/>
          <s:element minOccurs="1" maxOccurs="1" name="lastModified" nillable="true"
```

```
type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1" name="solutionDate" nillable="true"
type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="customer" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="index" nillable="true"
type="s:long" />
          <s:element minOccurs="0" maxOccurs="1" name="values" type="s0:ArrayOfKV"
/>
          <s:element minOccurs="0" maxOccurs="1" name="attachments"
type="s0:ArrayOfAttachment" />
          <s:element minOccurs="0" maxOccurs="1" name="body" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="TaskItem">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="type" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="queue" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="queueName" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="responsible" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="status" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="rowVersion" type="s:string"
/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfKV">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="KV" nillable="true"
type="s0:KV" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="KV">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="k" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="v" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfAttachment">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Attachment"
nillable="true" type="s0:Attachment" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="Attachment">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="filename" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="creator" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="created" nillable="true"
type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="ext" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="size" type="s:long" />
          <s:element minOccurs="0" maxOccurs="1" name="path" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="data" type="s:base64Binary"
/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfSkill">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Skill"
nillable="true" type="s0:Skill" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="Skill">
        <s:sequence>
```

```
          <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="value" type="s:int" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="RequiredAgents">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="expiryTime" nillable="true"
type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="agents"
type="s0:ArrayOfRequiredAgent" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfRequiredAgent">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="RequiredAgent"
nillable="true" type="s0:RequiredAgent" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="RequiredAgent">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="login" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="CreateTaskResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="response" nillable="true"
type="s0:TaskItem" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetTaskList">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="top" nillable="true"
type="s:int" />
            <s:element minOccurs="0" maxOccurs="1" name="modified" type="s:string"
/>
            <s:element minOccurs="0" maxOccurs="1" name="item" type="s0:TaskItem" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetTaskListResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="response" nillable="true"
type="s0:GetTaskListResult" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="GetTaskListResult">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="serverDate" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="items"
type="s0:ArrayOfTaskItem" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfTaskItem">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="TaskItem"
nillable="true" type="s0:TaskItem" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetTaskContent">
```

```
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="id" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="filter" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetTaskContentResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="response" nillable="true"
type="s0:TaskData" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="UpdateTask">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="data" type="s0:TaskData" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="UpdateTaskResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="response" nillable="true"
type="s0:TaskItem" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="CreateTaskSoapIn">
    <wsdl:part name="parameters" element="s0:CreateTask" />
  </wsdl:message>
  <wsdl:message name="CreateTaskSoapOut">
    <wsdl:part name="parameters" element="s0:CreateTaskResponse" />
  </wsdl:message>
  <wsdl:message name="GetTaskListSoapIn">
    <wsdl:part name="parameters" element="s0:GetTaskList" />
  </wsdl:message>
  <wsdl:message name="GetTaskListSoapOut">
    <wsdl:part name="parameters" element="s0:GetTaskListResponse" />
  </wsdl:message>
  <wsdl:message name="GetTaskContentSoapIn">
    <wsdl:part name="parameters" element="s0:GetTaskContent" />
  </wsdl:message>
  <wsdl:message name="GetTaskContentSoapOut">
    <wsdl:part name="parameters" element="s0:GetTaskContentResponse" />
  </wsdl:message>
  <wsdl:message name="UpdateTaskSoapIn">
    <wsdl:part name="parameters" element="s0:UpdateTask" />
  </wsdl:message>
  <wsdl:message name="UpdateTaskSoapOut">
    <wsdl:part name="parameters" element="s0:UpdateTaskResponse" />
  </wsdl:message>
  <wsdl:portType name="TMISoap">
    <wsdl:operation name="CreateTask">
      <wsdl:input message="tns:CreateTaskSoapIn" />
      <wsdl:output message="tns:CreateTaskSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetTaskList">
      <wsdl:input message="tns:GetTaskListSoapIn" />
      <wsdl:output message="tns:GetTaskListSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetTaskContent">
```

```
      <wsdl:input message="tns:GetTaskContentSoapIn" />
      <wsdl:output message="tns:GetTaskContentSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="UpdateTask">
      <wsdl:input message="tns:UpdateTaskSoapIn" />
      <wsdl:output message="tns:UpdateTaskSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="TMISoap" type="tns:TMISoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="CreateTask">
      <soap:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetTaskList">
      <soap:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetTaskContent">
      <soap:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateTask">
      <soap:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="TMISoap12" type="tns:TMISoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="CreateTask">
      <soap12:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetTaskList">
      <soap12:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
```

```
    </wsdl:operation>
    <wsdl:operation name="GetTaskContent">
      <soap12:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateTask">
      <soap12:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="TMI">
    <wsdl:port name="TMISoap" binding="tns:TMISoap">
      <soap:address location="http://10.31.96.65:1080/TMI/TMI.asmx" />
    </wsdl:port>
    <wsdl:port name="TMISoap12" binding="tns:TMISoap12">
      <soap12:address location="http://10.31.96.65:1080/TMI/TMI.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

9 ScriptResultService Specification

ScriptResultService contains the following operations:

- GetScriptResults => Retrieves script results from a task.
- SetScriptResult => Sets script result for a task.

Each operation consists of two messages: the request and the response.
If a Sinch Contact Pro user is authenticated, then queue right MANAGE_CONTACTS (Manage Contact History) is checked when retrieving or setting script results. User does not need to have right to script, but instead user needs to have right to queue, and then user gets right to that script that is defined for the queue. Also if user has right to queue Q1 which has script S1, and user has right to queue Q2 which has script S2, then user also has right to create script S1 for queue Q2 and script S2 for queue Q1.

9.1 GetScriptResults Operation

9.1.1 Definition

Purpose	Retrieves script results from a task (or e-mail).
Request name	GetScriptResults
Request elements	String <contactId> => Task or e-mail Id (GUID). The user needs to have right to the queue in which this task currently is. String <scriptNameFilter> => Optional filter for scriptName. By default all scripts (that the user has right to see) are returned. This parameter is used in SQL LIKE operand and the pattern can contain % wildchar character to match any characters.
Response name	GetScriptResultsResponse
Response elements	ScriptResult array <response> => Array of ScriptResult. Each ScriptResult has the following sub-elements: String <contactId> => GUID. This identifies the task/e-mail. String <customerId> => GUID. This identifies the outbound customer. This is not supported currently; value will be null always. String <scriptName> => Name of the script Float <sumAmount> => Sum of amount values in answers, or 0. DateTime <modified> => When this script result was last modified (in UTC).

	ScriptResultDetail array <details> => Array of ScriptResultDetail. Each ScriptResultDetail has the following sub-elements: String <question> => Question. String <answer> => Answer, either from script definition or free text, depending on controlType. Float <amount > => Amount, if question allows a separate amount value in addition to answer. Int <ordinal> => Ordinal for script result detail. This is not the same as question ordinal. Ordinals start from 0. String <controlType> => Control type, one of: RADIOBUTTON, CHECKBOX, DROPDOWNLIST, COMBOBOX, INPUTTEXT, TEXTAREA, DROPDOWNCHECKBOX String <dataType> => Data type for answer, one of: TEXT, INTEGER, FLOATING, DATETIME, CURRENCY DateTime <modified> => When this script result detail was last modified (in UTC).

9.1.2 GetScriptResults Message

This message carries a GetScriptResults operation request and its parameters from a SOAP client to ScriptResultService.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetScriptResults xmlns="urn:TMI">
      <contactId>616B5EA2A74CE111A0560003FF38AC85</contactId>
      <scriptNameFilter></scriptNameFilter>
    </GetScriptResults>
  </soap:Body>
</soap:Envelope>
```

9.1.3 GetScriptResultsResponse Message

This message carries results of a GetScriptResults operation from ScriptResultService back to a SOAP client.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetScriptResultsResponse xmlns="urn:TMI">
      <response>
```

```
        <contactId>616B5EA2A74CE111A0560003FF38AC85</contactId>
        <customerId></customerId>
        <scriptName>Interactive Test</scriptName>
        <sumAmount>0</sumAmount>
        <modified>2012-02-02T12:13:57.227</modified>
        <details>
          <ScriptResultDetail>
            <question>Second Q: How old are you?</question>
            <answer>31-60</answer>
            <amount>0</amount>
            <ordinal>0</ordinal>
            <controlType>RADIOBUTTON</controlType>
            <modified>2012-02-03T10:28:06.633</modified>
          </ScriptResultDetail>
          <ScriptResultDetail>
            <question>Third Q: What next?</question>
            <answer>don't know</answer>
            <amount>0</amount>
            <ordinal>1</ordinal>
            <controlType>INPUTTEXT</controlType>
            <dataType>TEXT</dataType>
            <modified>2012-02-03T10:28:07.123</modified>
          </ScriptResultDetail>
        </details>
      </response>
    </GetScriptResultsResponse>
  </soap:Body>
</soap:Envelope>
```

9.2 SetScriptResult Operation

9.2.1 Definition

Purpose	Sets script result for a task (or e-mail).
Request name	SetScriptResult
Request elements	ScriptResult <scriptResult> => ScriptResult which has the following sub-elements: String <contactId> => GUID. This identifies the task/e-mail. String <customerId> => GUID. This identifies the outbound customer. This is not supported currently, value should be null always. String <scriptName> => Name of the script. Must be a valid Script.Name from database. Float <sumAmount> => Ignored in input, calculated from details or answers. DateTime <modified> => Ignored in input, the current time (in UTC). ScriptResultDetail array <details> => Array of ScriptResultDetail. Each ScriptResultDetail has the following sub-elements: String <question> => Question. Should be either existing ScriptQuestion.Data or ScriptQuestion.Ordinal. Will be always ScriptQuestion.Data in output. Question ordinal can be used to identify the question in input, so that question text can change later.

	String <answer> => Answer, either from script definition (ScriptAnswer.Data) or free text, depending on controlType. Float <amount > => Amount, if question allows a separate amount value in addition to answer. Int <ordinal> => Ignored in input. Ordinal is generated so that first ScriptResultDetail gets 0, the next 1, etc. String <controlType> => Ignored in input. Control type is taken from ScriptQuestion.ControlType, one of: RADIOBUTTON, CHECKBOX, DROPDOWNLIST, COMBOBOX, INPUTTEXT, TEXTAREA, DROPDOWNCHECKBOX Free text in answer is allowed only if control type is one of COMBOBOX, INPUTTEXT, TEXTAREA. In other cases answer must match existing ScriptAnswer.Data. String <dataType> => Ignored in input. Data type is taken from ScriptQuestion.DataType, one of: TEXT, INTEGER, FLOATING, DATETIME, CURRENCY DateTime <modified> => Ignored in input. When this script result detail was last modified (in UTC).
Response name	SetScriptResultResponse
Response elements	Empty. Call GetScriptResults to get the script result back.

9.2.2 SetScriptResult Message

This message carries a SetScriptResult operation request and its parameters from a SOAP client to ScriptResultService. In the example below, all elements that should have no value, like <modified>, could be given as null values (xsi:nil="true") but they are not shown here for simplicity.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetScriptResult xmlns="urn:TMI">
      <scriptResult>
        <contactId>616B5EA2A74CE111A0560003FF38AC85</contactId>
        <scriptName>Interactive Test</scriptName>
        <details>
          <ScriptResultDetail>
            <question>Second Q: How old are you?</question>
            <answer>31-60</answer>
          </ScriptResultDetail>
          <ScriptResultDetail>
            <question>Third Q: What next?</question>
            <answer>don’t know</answer>
          </ScriptResultDetail>
        </details>
      </scriptResult>
    </SetScriptResult>
  </soap:Body>
</soap:Envelope>
```

9.2.3 SetScriptResultResponse Message

This message carries results of a SetScriptResult operation from ScriptResultService back to a SOAP client.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetScriptResultResponse xmlns="urn:TMI" />
  </soap:Body>
</soap:Envelope>
```

10 ScriptResultService Content Description

This section describes the contents of the XML elements that carry task information from a Sinch Contact Pro system via the interface.

10.1 GUID Formatting in Input and Output

IDs like ScriptResult.contactId are GUIDs internally, but Strings in SOAP interface. GUID format is UPPERCASE and without - characters, so exactly 32 characters, like `C4ADBBB108F7DF11B65A0003FF38AC85` in output always. In input to ScriptResultService interface, the hexadecimal characters can be in lowercase also, and there can be - characters, so the following format is accepted as well: `c4adbbb1-08f7-df11-b65a-0003ff38ac85`

10.2 ScriptResult Element

10.2.1 Definition

Purpose	The element contains the task and script identification and has a collection of ScriptResultDetails for a task.
Element name	ScriptResult
Scope	GetScriptResults, SetScriptResult
Sub-elements	String <contactId> => GUID, identifies the task. String <contactId> => GUID. This identifies the task/e-mail. String <customerId> => GUID. This identifies the outbound customer. This is not supported currently, value should be null always. String <scriptName> => Name of the script. Must be a valid Script.Name from database. Float <sumAmount> => Ignored in input, calculated from details or answers. DateTime <modified> => Ignored in input, the last modification time (in UTC). ScriptResultDetail array <details> => Array of ScriptResultDetail.

10.2.2 XML Schema Description

```
<s:complexType name="ScriptResult">
   <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="contactId" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="customerId" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="scriptName" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="sumAmount" nillable="true"
type="s:float" />
      <s:element minOccurs="1" maxOccurs="1" name="modified" nillable="true"
type="s:dateTime" />
      <s:element minOccurs="0" maxOccurs="1" name="details"
type="s0:ArrayOfScriptResultDetail" />
   </s:sequence>
</s:complexType>
```

10.3 ScriptResultDetail Element

10.3.1 Definition

Purpose	The element contains the question – answer pair.
Element name	ScriptResultDetail
Scope	GetScriptResults, SetScriptResult
Sub-elements	String <question> => Question. String <answer> => Answer, either from script definition (ScriptAnswer.Data) or free text, depending on controlType. Float <amount > => Amount, if question allows a separate amount value in addition to answer. Int <ordinal> => Ignored in input. Ordinal is generated so that first ScriptResultDetail gets 0, the next 1, etc. String <controlType> => Ignored in input. Control type is taken from ScriptQuestion.ControlType, one of: RADIOBUTTON, CHECKBOX, DROPDOWNLIST, COMBOBOX, INPUTTEXT, TEXTAREA, DROPDOWNCHECKBOX Free text in answer is allowed only if control type is one of COMBOBOX, INPUTTEXT, TEXTAREA. In other cases answer must match existing ScriptAnswer.Data. String <dataType> => Ignored in input. Data type is taken from ScriptQuestion.DataType, one of: TEXT, INTEGER, FLOATING, DATETIME, CURRENCY DateTime <modified> => Ignored in input. When this script result detail was last modified (in UTC).

10.3.2 XML Schema Description

```
<s:complexType name="ScriptResultDetail">
   <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="question" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="answer" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="amount" nillable="true"
type="s:float" />
      <s:element minOccurs="1" maxOccurs="1" name="ordinal" nillable="true"
type="s:int" />
      <s:element minOccurs="0" maxOccurs="1" name="controlType" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="dataType" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="modified" nillable="true"
type="s:dateTime" />
   </s:sequence>
</s:complexType>
```

11 ScriptResultService WSDL

Description

The WSDL (http://www.w3.org/TR/wsdl) description of the interface is the following:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s0="urn:TMI"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://sap.com/bcm/TMI"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://sap.com/bcm/TMI"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="urn:TMI">
      <s:element name="GetScriptResults">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="contactId" type="s:string"
/>
            <s:element minOccurs="0" maxOccurs="1" name="scriptNameFilter"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetScriptResultsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="unbounded" name="response"
nillable="true" type="s0:ScriptResult" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ScriptResult">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="contactId" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="customerId" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="scriptName" type="s:string"
/>
          <s:element minOccurs="1" maxOccurs="1" name="sumAmount" nillable="true"
type="s:float" />
          <s:element minOccurs="1" maxOccurs="1" name="modified" nillable="true"
type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="details"
type="s0:ArrayOfScriptResultDetail" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfScriptResultDetail">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="ScriptResultDetail"
nillable="true" type="s0:ScriptResultDetail" />
        </s:sequence>
```

```
      </s:complexType>
      <s:complexType name="ScriptResultDetail">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="question" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="answer" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="amount" nillable="true"
type="s:float" />
          <s:element minOccurs="1" maxOccurs="1" name="ordinal" nillable="true"
type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="controlType" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="dataType" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="modified" nillable="true"
type="s:dateTime" />
        </s:sequence>
      </s:complexType>
      <s:element name="SetScriptResult">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="scriptResult"
type="s0:ScriptResult" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="SetScriptResultResponse">
        <s:complexType />
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetScriptResultsSoapIn">
    <wsdl:part name="parameters" element="s0:GetScriptResults" />
  </wsdl:message>
  <wsdl:message name="GetScriptResultsSoapOut">
    <wsdl:part name="parameters" element="s0:GetScriptResultsResponse" />
  </wsdl:message>
  <wsdl:message name="SetScriptResultSoapIn">
    <wsdl:part name="parameters" element="s0:SetScriptResult" />
  </wsdl:message>
  <wsdl:message name="SetScriptResultSoapOut">
    <wsdl:part name="parameters" element="s0:SetScriptResultResponse" />
  </wsdl:message>
  <wsdl:portType name="ScriptResultServiceSoap">
    <wsdl:operation name="GetScriptResults">
      <wsdl:input message="tns:GetScriptResultsSoapIn" />
      <wsdl:output message="tns:GetScriptResultsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="SetScriptResult">
      <wsdl:input message="tns:SetScriptResultSoapIn" />
      <wsdl:output message="tns:SetScriptResultSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ScriptResultServiceSoap" type="tns:ScriptResultServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc" />
    <wsdl:operation name="GetScriptResults">
      <soap:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetScriptResult">
      <soap:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
```

```
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="ScriptResultServiceSoap12" type="tns:ScriptResultServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc" />
    <wsdl:operation name="GetScriptResults">
      <soap12:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetScriptResult">
      <soap12:operation soapAction="http://sap.com/bcm/TMI" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ScriptResultService">
    <wsdl:port name="ScriptResultServiceSoap" binding="tns:ScriptResultServiceSoap">
      <soap:address location="http://10.31.96.65:1080/TMI/ScriptResultService.asmx"
/>
    </wsdl:port>
    <wsdl:port name="ScriptResultServiceSoap12"
binding="tns:ScriptResultServiceSoap12">
      <soap12:address
location="http://10.31.96.65:1080/TMI/ScriptResultService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

12 Troubleshooting

The TMI log file is in the same directory as other Virtual Unit log files (for example, C:\SAP\BCM\VU1\logs) and its name has the following syntax: TMI_VU1_20050105.log.
The beginning of the file contains version and configuration information:

```
--------------------
2011-08-17  (Wednesday, 17 August 2011)
SAP BCM TMI 7.0.1.10033 1.0.4203.764 2011-07-05 01:25:30Z
file:///D:/Program Files/SAP/BCM/VU/VU_OII/web/TMI/bin/WicomTMI.DLL
Domain /LM/W3SVC/3/ROOT/TMI-4-129580499020836535 with ID = 5.
Process id=4464 name=w3wp working set=102MB private=85MB handles=1366 tot
cpu=00:23:20.7656250 user cpu=00:20:09.8125000
VU=VU_OII UseSecureLogging=False TickCount=373170203
LOG_DETECT_REPEATED = 1
LOG_KEEP_DAYS = 7
LOG_LEVEL = 3
LOG_SHOW_THREAD = 1
----ENV START---- 40
ALLUSERSPROFILE=C:\ProgramData
…
----ENV END----

LOG STATS: EXC:0 ERR:0 INF:2 TRC:0 DBG:0
10:18:27.468 INF> AppState: Uninitialized => Stopping
10:18:27.468 INF> Stop... reason=Start
…
10:18:27.566 TRC> 151 Users in 16 ms
…
10:18:27.578 INF> 8 Skills in 0 ms
10:18:27.578 INF> AppState: Stopping => Initialized
10:18:27.579 TRC> --- SOAPAction=  Remote=10.66.193.43  URL=/TMI/TMI.asmx
10:18:27.582 INF> (Status) StatusChecker started
10:18:27.884 TRC> (Status) State=Initialized  Users=151 Queues=33
…
```

Each line after that contains the time, the debugging level, the thread name, and the actual log information. The EXC> and ERR> debugging levels are exceptions and errors which should not happen during normal operation.

```
10:18:27.884 TRC> (Status) State=Initialized  Users=151 Queues=33
```

The example line above contains the TMI status and it is written once a minute.

- State= Initialized => Indicates that the interface is in working condition.
- Users=151 => Indicates that there are 151 users currently cached in TMI memory.
- Queues=33 => Indicates that currently there are 33 queues cached in TMI memory.

13 Glossary

Term	Description
CCtr	Sinch Contact Pro, formely SAP Contact Center
CEM	Communication Event Manager server that handles phone calls and e-mail tasks
CRM	Customer Relationship Management
ECF	Embedded Communications Framework
IA	Sinch Contact Pro Infrastructure Administrator
IIS	Microsoft Internet Information Services
SOAP	Simple Object Access Protocol
TMI	Task Management Interface
XRI	External Routing Item, task type used with ECF