

ITML  
PROVISIONING  
SPECIFICATION  
WORKING DRAFT



Last Updated: 02/19/2001

Control Number:

Version: 0.82

Editor(s): David Orchard

Copyright (C) Jamcracker, Inc.  
All Rights Reserved

Jamcracker, Inc.

Revisions

ITML PROVISIONING .....	1
ABSTRACT .....	1
STATUS OF THIS DOCUMENT .....	1
SPECIFICATION STATUS .....	1
RELATIONSHIP TO OTHER STANDARDS .....	1
<i>Normative</i> .....	2
<i>Non-Normative</i> .....	2
AUDIENCE .....	2
DOCUMENT CONVENTIONS.....	2
USE CASES.....	3
ACTOR .....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<i>Add User</i> .....	3
<i>Add Contact to CRM system</i> .....	3
<i>Modify User</i> .....	3
<i>Disable User</i> .....	3
<i>Enable User</i> .....	3
<i>Delete User</i> .....	3
ORGANIZATIONS.....	3
<i>Add Company</i> .....	3
<i>Modify Company</i> .....	4
<i>Disable Company</i> .....	4
<i>Enable Company</i> .....	4
<i>Delete Company</i> .....	4
SERVICE.....	4
<i>Subscribe Service for Company</i> .....	4
<i>Activate Service for Company</i> .....	4
<i>Upgrade/Downgrade Service for Company</i> .....	4
<i>Suspend Service for Company</i> .....	4
<i>Enable Service for Company</i> .....	4
<i>Delete Service for Company</i> .....	4
<i>Order a Service for User(s)</i> .....	4
<i>Upgrade/Downgrade a Service for User(s)</i> .....	4
<i>Suspend a Service for User(s)</i> .....	4
<i>Enable a Service for User(s)</i> .....	4
<i>Delete a Service for User(s)</i> .....	4
REQUIREMENTS .....	5
USER PROVISIONING MESSAGES .....	6
NAMESPACE .....	6
REQUESTS.....	6
ERRORS .....	6
PROVISIONING SCHEMAS INFORMATION .....	7
PROVISIONING STATES AND STATE TRANSITIONS .....	7
PROVISIONING DATA MODE .....	7
<i>Users, roles and credentials</i> .....	8
<i>Accounts and Contacts</i> .....	9
TRANSACTIONIDS .....	9
EFFECTIVE DATING .....	10
PROVISIONING DOCUMENTS .....	11
SCHEMA FOR PROVISIONING COMMANDS – ITMLPROVMETHODS.XSD.....	12
SCHEMA FOR PROVISIONING STRUCTURES – ITMLPROVSTRUCTS.XSD.....	15

ITMLADDCOMPANY.XML SAMPLE DOCUMENT .....	20
ITMLADDCOMPANYRESPONSE.XML SAMPLE DOCUMENT .....	21
CONFORMANCE .....	22
APPENDIX .....	23
ADDITIONAL NORMATIVE DOCUMENTS .....	23
ISSUES .....	23
TYPICAL INFORMATION FOR EACH PARTNER (NON-NORMATIVE).....	23
FUTURES (NON-NORMATIVE) .....	24
PARTNER GUIDELINES (NON-NORMATIVE) .....	24
CREDITS .....	24
REVISION HISTORY .....	24

---

# ITML Provisioning

## Abstract

ITML - the Information Technology Markup Language - is a set of specifications of protocols, message formats and best practices in the ASP and ASP aggregation market to provide seamless integration of partners and business processes. It is based upon open-standards, particularly XML and HTTP. It is an implementation of the ITML Messaging and Protocol Specification.

ITML provisioning specifies a set of protocols and documents that Jamcracker and its partners exchange that provides for secure and reliable user and company information synchronization. Users, companies, services for users, and services for companies can be added, modified, disabled, enabled or deleted across the set of Jamcracker and it's partners. Jamcracker or its partners can create UserID, Password, and CompanyID information in the provisioning conversations.

This specification describes the following key decisions:

- use cases
- requirements
- information model including data model and state transitions
- provisioning requests schemas and samples
- provisioning responses schemas and samples including errors
- provisioning namespace

This specification provides a framework for a one way data transfer between Jamcracker and the partner. Partners that wish to act in roles other than recipient of data will use a different specification than this.

## Status of this Document

This document is a Working Draft, issued by the Jamcracker ITML team, for review by selected partners. It is not intended for dissemination beyond direct recipients.

The ITML team expects that significant changes will occur in this document before version 1.0 is released. The ITML Team will not allow early implementation to constrain its ability to make changes to this specification prior to final release.

## Specification Status

This specification is incomplete in some regards. The samples and schemas are fragments only, without the SOAP enveloping information. The use of the SOAP Schema must be integrated with the samples and schemas. This is delayed because of the lack of multiple namespaces in authoring tools, specifically XML Spy

## Relationship to other standards

The ITML User Provisioning specification has been influenced by many recent standards efforts including, but not limited to, the following:

## **Normative**

ITML Messaging and Protocol Specification

## **Non-Normative**

ISO Language codes

<http://www.oasis-open.org/cover/iso639a.html>

## **Audience**

This document is a technical specification and is intended for developers and architects.

## **Document Conventions**

The following notations are used to present material in this document:

**ISSUE:** An issue is a direct request for feedback from the audience. An issue reflects a lack of decision due to insufficient or conflicting inputs. These are resolved through the acquisition of more input

**NOTE:** Extra normative information that the author(s) wish to draw the attention of the reader to.

---

## Use Cases

The following Use cases describe the interactions supported by this specification. In many cases, the content of the use case follows a straight-forward template. In these cases, only the name of the use case is shown and not the content. The template is

1. Update an entity(s) state information at Jamcracker platform,
2. Send state change to partner
3. Partner returns success or failure

**ISSUE:** Should some or all use cases be optional. In particular, Modify and Delete user. Imagine where 1 Partner supports delete but another only supports disable. This potentially leaves the distributed repository in an invalid state. Author recommendation #1: Require all systems to support the various state transitions. For example, if a system does not support disable/enable user but does support delete, that is functionality that is presumably desirable. Is there significant enough effort required to add an enable/disable bit to a db table?

## User

### Add User

1. A user is added at the Jamcracker workspace.
2. Add User request is sent to Partner. This contains a userID and password assigned by Jamcracker. The user request contains a companyID for the User's company that has previously been defined.
3. The userID and Password accepted by the partner is returned. Different values than those sent by Jamcracker indicate that the partner has its own assignment mechanism.

### Add Contact to CRM system

1. A contact is added at the Jamcracker workspace. This is not a end-user
2. Add User request is sent to Partner, with CRM Contact as the parameter.
3. Success or Failure is returned

### Modify User

1. User information is modified at the Jamcracker workspace. This can include password change
2. Update User request with User information is sent to Partner.
3. An empty body or failure is returned by partner

### Disable User

### Enable User

### Delete User

## Organizations

### Add Company

1. A company is added at the Jamcracker workspace.
2. Create company request is sent to Partner. This contains a companyID created by Jamcracker.

3. The companyID accepted by the partner is returned. Different values than those sent by Jamcracker indicate that the partner has its own assignment mechanism.

### **Modify Company**

### **Disable Company**

### **Enable Company**

### **Delete Company**

### **Service**

### **Subscribe Service for Company**

### **Activate Service for Company**

### **Upgrade/Downgrade Service for Company**

### **Suspend Service for Company**

### **Enable Service for Company**

### **Delete Service for Company**

### **Order a Service for User(s)**

### **Upgrade/Downgrade a Service for User(s)**

### **Suspend a Service for User(s)**

### **Enable a Service for User(s)**

### **Delete a Service for User(s)**

**Note:** Jamcracker sends a fully populated request – ie user or company – in each request. The Partner may ignore the content it is not interested in. In the extreme, the partner may get requests to change data that it does not store.

---

## Requirements

The following requirements are met by this specification

1. User additions and modifications can be made to a Partner with no human intervention on the Jamcracker or Partner side.
2. Company additions and modifications can be made to a Partner with no human intervention on the Jamcracker or Partner side.
3. Services for User additions and modifications can be made to a Partner with no human intervention on the Jamcracker or Partner side.
4. Services for Company additions and modifications can be made to a Partner with no human intervention on the Jamcracker or Partner side.
5. User Ids can be created by Partners or Jamcracker
6. Company Ids can be created by Partners or Jamcracker.
7. Users may have multiple credentials associated with a userID.
8. **ISSUE:** Should passwords be sent or not?

## Design Goals

1. ITML provisioning uses industry standard technologies.
2. The set of elements and operations shall be as small as possible
3. The elements will be globalization ready
4. The elements and operations will be easily extensible



---

# User Provisioning Messages

## Namespace

All Provisioning elements are associated with the prov namespace identifier, which is bound to the namespace URI <http://www.itml.org/ns/provisioning/>. Errors are in the prov-err namespace identifier, bound to <http://www.itml.org/ns/provisioning/err>.

## Requests

The provisioning specification consists of a number of commands and responses. The XML Schema for the commands and responses, as well as many samples, are provided throughout this document. The following request types with parameters are provided:

- AddUser(UserInfo) – Add a user with the provided Data, returns UserID and password
- UpdateUser(UserID, UserInfo) – Update user data to provided Data
- UpdateUser(UserID, Status) – Render user account inactive or active. .
- UpdateUser(UserID, ServiceID, Status) - Add/Disable/Enable/Delete service for user
- UpdateUser(UserID, ServiceID, ServiceID, ServiceStatus ) - upgrade/downgrade service for user.
- DeleteUser(UserID) – Delete a user account.
- AddCompany(Company) – add a company with the provided data, returns companyID accepted by the partner
- UpdateCompany(CompanyID, CompanyInfo)
- UpdateCompany(CompanyID, Status) – Render company inactive or active
- UpdateCompany(CompanyID, ServiceID, Status) – Add/Activate/Disable/Enable/Delete services for company
- DeleteCompany(CompanyID) – Delete a company account.

**NOTE:** There is no accepted standard for defining interfaces to XML/Web services. It is expected that this specification will evolve to any standard that emerges, such as NASSL, WSDL, SDL, etc.

## Errors

The following error conditions are defined in the prov-err namespace:

- InvalidUserID
- InvalidUserInfo
- DuplicateUserID
- UserNotDisabled – when an attempt to enable a non-disabled user occurs
- UserNotActive – User has been deleted so update commands are erroneous
- InvalidCompanyID
- InvalidCompanyInfo
- DuplicateCompanyID
- CompanyNotDisabled - when an attempt to enable a non-disabled user occurs
- CompanyNotActive – Company has been deleted so update commands are erroneous

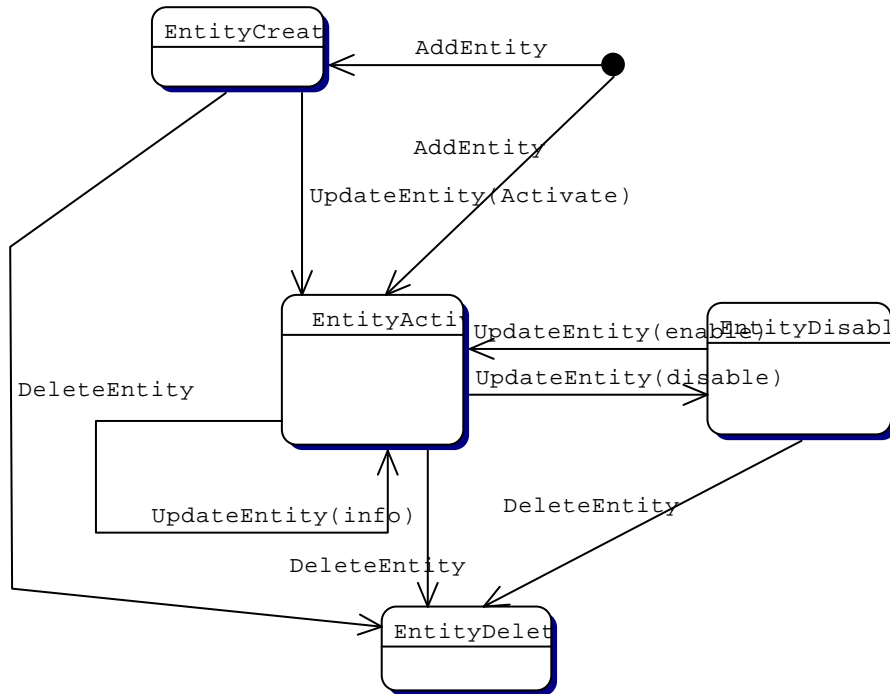
---

## Provisioning Schemas Information

This section provides additional information above and beyond the provisioning schemas and documents.

## Provisioning States and State Transitions

The state transition model for entities in ITML Provisioning, specifically Actors, Companies and Services is:



STD showing the general state transitions of entities in the system, such as Users, companies and services

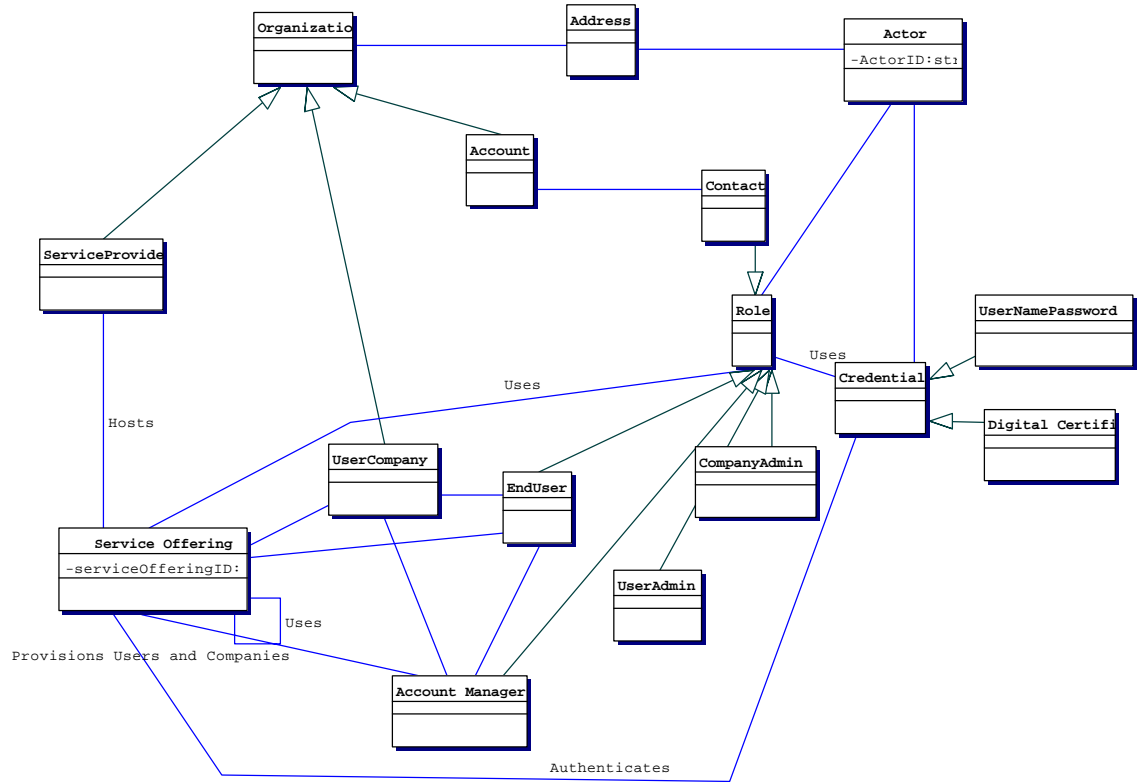
Add and Delete verbs are individual events

Modify information and modify state to activate, enable or disable are bundled into update events

All entities do not have to correspond fully to this STD. As an example, services have the ServiceCreated state, but users and companies do not. All 3 have the Active, Inactive and deleted states.

## Provisioning Data Model

The data model for provisioning follows:



## Users, roles and credentials

Users may have:

- 1. Multiple Authentication/Credential types - username/password is most common
- 2. Multiple Roles per Credential.
- 3. Multiple Credentials per Role. This allows multiple credentials for lower level roles.

Some Examples:

- 1. User has ID/pwd X/Y for roles EndUser and UserAdmin
- 2. User has x.509 cert X for role UserAdmin and ID/pwd a/b for role EndUser
- 3. User has x.509 cert X for role EndUser and role UserAdmin, and ID/pwd a/b for role User.

Roles are defined in terms of what a user can do with Provisioning defined data types. Users and Companies are defined in provisioning and are thus maintainable by provisioning commands. EndUsers are users who are allowed to access a partner system. Authorization above and beyond this is defined by the partner. UserAdmin and CompanyAdmin are users who can modify user or company information – as defined by this specification. Users can administer their own User information. A CompanyAdmin should not modify User information, nor should a UserAdmin modify company information. Effectively, the UserAdmin or CompanyAdmin can cause User or Company add/update/delete commands to occur.

A Partner may apply further classification information in their specific set of information. The enumeration value "PartnerDefined" indicates that a partner has defined their own particular role(s) and the specific values for the roles are included in the partner specification extensions.

Where a partner does not support aspects of this data model, then Jamcracker will define the authentication appropriate for a given request and whether multiple user identities need to be managed. This allows multiple credentials to be used for a given user, typically for different client access locations – mobile, kiosk, fixed pc.

For example, partner M may support only 1 username per user and partner E supports multiple credentials per user, then Jamcracker may define that the UserID U is defined once in JC yet twice in Partner M. JC would then track that updates to UserID U require multiple updates to partner M.

Jamcracker requires that the namespace for usernames, userIDs and companyIDs be completely addressable by Jamcracker. There are no values that are excluded to Jamcracker's use.

**ISSUE:** Is this valid? An issue arises if each partner controls a different portion of the namespace, and the mapping that may occur.

## **Accounts and Contacts**

Accounts are companies that are stored that do not have end users. Contacts are people that are not end users. These are for CRM and SFA systems where a partner is tracking companies and contacts for the purpose of generating sales. Thus the contact has addresses and names, but not credentials.

**ISSUE:** [from OpenAir] if names, passwords get modified during the user creation process this will create problems if a user tries to log-in directly to the application and not through the portal (often wireless or disconnected clients need to authenticate directly to the application site)

**ISSUE:** User Passwords sent to Partner? Due to security concerns about sending passwords, as well as emerging standardization around Single Sign-on, user passwords are not sent to partners as part of user provisioning.

## **TransactionIDs**

Every ITML request contains a transaction ID. This is a value of the originating companyID followed by a 4 byte value. The transaction ID is included in the request

There is mention of transaction IDs in SOAP. However, it is unclear how these are to be used, created and conformance tested. Thus the decision to use an attribute in the body rather than an element in the header.

## **ITML Provisioning Encoding rules**

Actors are always sent with address, name, role and credential information. Actors never contain service information.

Organizations are always sent with address information. Companies never contain service information

Collections of Services for a User are not supported. This would cause a transaction problem if there was an error with any particular service.

Collections of Users for a Company are not supported.

## Effective Dating

**ISSUE:** Oui ou Non for effective dating. Effective dating is a standard feature of mature ERP systems such as PeopleSoft. It does introduce complexity in many of the use cases due to overlaps or null-set intersections of the effective dates of multiple requests. For example, if a user is added effective a particular date, can a modify user with a prior effective date occur? Further, what is the expected behavior if some systems support effective dates and others don't.

Author recommendation: Don't support effective dates in version 1.0 because of lack of compliance, interoperability and increased complexity.

From Employease comments 2000 09 13"retroactive modifications to data MUST be supported. In addition, error correction of epochs MUST be supported. If these requirements are not supported, then a partner cannot synchronize the data via an automated channel. New Issue - we need to establish a business rule as to how the partner applications will deal with a historical sequence of records.

New Issue - What is the transaction boundary inside the data set? At what level do data elements need an effective date?

need to define history rules , suggestion:

- a) if an epoch start date is transmitted that is previous to a record effective date, the future records are deleted. Alternatively, a start/stop date for an epoch could be transmitted.
- b) if an epoch start date is transmitted that is between an existing record start/terminate date, the existing record will be truncated, and linked to the new record.

“

Sample Use case for illustration:

Update User Information A with effective date in the future received by Jamcracker. ASP E supports effective dates and ASP Y does not support effective dates. JC has 2 options: 1) pass request through to E with effective date, and hold request for Y until effective date reached. 2) hold requests for both E and Y until effective date reached then pass request.

Possible resolutions:

1. Drop effective dating from specification
2. Add effective dating with only start date
3. Add effective dating with start and stop date.

---

## **Provisioning Documents**

The following sections contain the provisioning Schemas and instances of provisioning documentation

---

## Schema for Provisioning commands – ITMLProvMethods.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema" elementFormDefault="qualified">
  <xsd:include schemaLocation="ITMLProvStructs.xsd"/>
  <xsd:element name="addUser">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="User" type="prov:UserType"/>
      </xsd:sequence>
      <xsd:attribute name="txid" type="prov:txidType"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="addUserResponse">
    <xsd:complexType>
      <xsd:choice>
        <xsd:sequence>
          <xsd:element name="UserIdentity" type="prov:UserIdentityType"/>
          <xsd:element name="UserNamePassword" type="prov:UserNamePasswordType"/>
          <xsd:element name="ProvisioningExtensions" type="prov:ProvisioningExtensionsType"
minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:element name="ITMLFaultDetail" type="prov:ITMLFaultDetailType" minOccurs="0"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="updateUser">
    <xsd:complexType>
      <xsd:choice>
        <xsd:sequence>
          <xsd:element name="UserIdentity" type="prov:UserIdentityType"/>
          <xsd:element name="Status" type="prov:StatusType"/>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element name="UserIdentity" type="prov:UserIdentityType"/>
          <xsd:element name="User" type="prov:UserType"/>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element name="UserIdentity" type="prov:UserIdentityType"/>
          <xsd:element name="ServiceID" type="prov:ServiceIDType"/>
          <xsd:element name="ServiceStatus" type="prov:ServiceStatusType"/>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element name="UserIdentity" type="prov:UserIdentityType"/>
          <xsd:element name="ServiceID" type="prov:ServiceIDType"/>
          <xsd:element name="ServiceID" type="prov:ServiceIDType"/>
          <xsd:element name="ServiceStatus" type="prov:ServiceStatusType"/>
        </xsd:sequence>
      </xsd:choice>
      <xsd:attribute name="txid" type="prov:txidType"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="updateUserResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ITMLFaultDetail" type="prov:ITMLFaultDetailType" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="deleteUser">
    <xsd:complexType>
      <xsd:sequence>
```

```

        <xsd:element name="UserIdentity" type="prov:UserIdentityType"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="deleteUserResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ITMLFaultDetail" type="prov:ITMLFaultDetailType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="addCompany">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Company" type="prov:CompanyType"/>
        </xsd:sequence>
        <xsd:attribute name="txid" type="prov:txidType"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="addCompanyResponse">
    <xsd:complexType>
        <xsd:choice>
            <xsd:sequence>
                <xsd:element name="CompanyID" type="prov:CompanyIDType"/>
                <xsd:element name="ProvisioningExtensions" type="prov:ProvisioningExtensionsType"
minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:element name="ITMLFaultDetail" type="prov:ITMLFaultDetailType" minOccurs="0"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
<xsd:element name="updateCompany">
    <xsd:complexType>
        <xsd:choice>
            <xsd:sequence>
                <xsd:element name="CompanyID" type="prov:CompanyIDType"/>
                <xsd:element name="Status" type="prov:StatusType"/>
            </xsd:sequence>
            <xsd:sequence>
                <xsd:element name="CompanyID" type="prov:CompanyIDType"/>
                <xsd:element name="Company" type="prov:CompanyType"/>
            </xsd:sequence>
            <xsd:sequence>
                <xsd:element name="CompanyID" type="prov:CompanyIDType"/>
                <xsd:element name="ServiceID" type="prov:ServiceIDType"/>
                <xsd:element name="ServiceStatus" type="prov:ServiceStatusType"/>
            </xsd:sequence>
        </xsd:choice>
        <xsd:attribute name="txid" type="prov:txidType"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="updateCompanyResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ITMLFaultDetail" type="prov:ITMLFaultDetailType" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="deleteCompany">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="CompanyID" type="prov:CompanyIDType"/>
        </xsd:sequence>
        <xsd:attribute name="txid" type="prov:txidType"/>
    </xsd:complexType>
</xsd:element>

```



```
<xsd:element name="deleteCompanyResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ITMLFaultDetail" type="prov:ITMLFaultDetailType" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

---

# Schema for Provisioning Structures – ITMLProvStructs.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema" elementFormDefault="qualified">
  <xsd:complexType name="AuthenticationTypeListType">
    <xsd:sequence>
      <xsd:element name="Role" type="prov:RoleType" minOccurs="0"/>
      <xsd:element name="UserNamePassword" type="prov:UserNamePasswordType"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="AuthenticationType" abstract="true">
    <xsd:sequence>
      <xsd:element name="Role" type="prov:RoleType" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="AddressType" abstract="true"/>
  <xsd:complexType name="FixedAddressType" abstract="true">
    <xsd:complexContent>
      <xsd:extension base="prov:AddressType">
        <xsd:sequence>
          <xsd:element name="AddressLocation" type="prov:AddressLocationType"/>
          <xsd:element name="CareOfName" type="xsd:string" nullable="true"/>
          <xsd:element name="AddressLine1" type="xsd:string"/>
          <xsd:element name="AddressLine2" type="xsd:string" nullable="true"/>
          <xsd:element name="Building" type="xsd:string"/>
          <xsd:element name="Country" type="xsd:string"/>
          <xsd:element name="Email" type="xsd:string"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="US-FixedAddressType">
    <xsd:complexContent>
      <xsd:extension base="prov:FixedAddressType">
        <xsd:sequence>
          <xsd:element name="County" type="xsd:string"/>
          <xsd:element name="POBox" type="prov:NA-POBoxType"/>
          <xsd:element name="State" type="prov:US-StateType"/>
          <xsd:element name="PostalCode" type="prov:US-PostalCodeType"/>
          <xsd:element name="Phone" type="prov:NA-PhoneNumberType"/>
          <xsd:element name="Fax" type="prov:NA-PhoneNumberType"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="CompanyAddressListType">
    <xsd:sequence>
      <xsd:element name="FixedAddress" type="prov:FixedAddressType" minOccurs="2" maxOccurs="2"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="MobileAddressType" abstract="true">
    <xsd:complexContent>
      <xsd:extension base="prov:AddressType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="US-MobileAddressType">
    <xsd:complexContent>
      <xsd:extension base="prov:MobileAddressType">
        <xsd:sequence>
          <xsd:element name="Phone" type="prov:NA-PhoneNumberType"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```

<xsd:complexType name="NameInfoType">
  <xsd:sequence>
    <xsd:element name="NamePrefix" type="xsd:string" nullable="true"/>
    <xsd:element name="FirstName" type="xsd:string"/>
    <xsd:element name="MiddleInitial" type="xsd:string" nullable="true"/>
    <xsd:element name="LastName" type="xsd:string"/>
    <xsd:element name="NameSuffix" type="xsd:string" nullable="true"/>
    <xsd:element name="KnownAs" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PreferencesType">
  <xsd:sequence>
    <xsd:element name="Language" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="RoleType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="UserAdmin"/>
    <xsd:enumeration value="PartnerDefined"/>
    <xsd:enumeration value="CompanyAdmin"/>
    <xsd:enumeration value="User"/>
    <xsd:enumeration value="Contact"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="UserType">
  <xsd:sequence>
    <xsd:element name="UserIdentity" type="prov:UserIdentityType"/>
    <xsd:element name="NameInfo" type="prov:NameInfoType"/>
    <xsd:element name="DateOfBirth" type="xsd:string"/>
    <xsd:element name="OrganizationInfo" type="prov:OrganizationInfoType"/>
    <xsd:element name="Gender" type="xsd:string"/>
    <xsd:element name="UserAddressList" type="prov:UserAddressListType"/>
    <xsd:element name="AuthenticationTypeList" type="prov:AuthenticationTypeListType"/>
    <xsd:element name="Preferences" type="prov:PreferencesType"/>
    <xsd:element name="OrganizationDates" type="prov:OrganizationDatesType"/>
    <xsd:element name="ProvisioningExtensions" type="prov:ProvisioningExtensionsType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ProvisioningExtensionsType">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="lax"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CompanyType">
  <xsd:sequence>
    <xsd:element name="CompanyID" type="prov:CompanyIDType"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="CompanyAddressList" type="prov:CompanyAddressListType"/>
    <xsd:element name="ProvisioningExtensions" type="prov:ProvisioningExtensionsType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UserAddressListType">
  <xsd:sequence>
    <xsd:element name="FixedAddress" type="prov:US-FixedAddressType"/>
    <xsd:element name="MobileAddress" type="prov:US-MobileAddressType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UserIDType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="IsDefinedByPartner" type="xsd:boolean"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

```

<xsd:complexType name="CompanyIDType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="IsDefinedByPartner" type="xsd:boolean"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="ServiceIDType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="StatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Modify"/>
    <xsd:enumeration value="Disable"/>
    <xsd:enumeration value="Enable"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ServiceStatusType">
  <xsd:restriction base="prov:StatusType">
    <xsd:enumeration value="Add"/>
    <xsd:enumeration value="Activate"/>
    <xsd:enumeration value="Delete"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="UserIdentityType">
  <xsd:sequence>
    <xsd:element name="UserID" type="prov:UserIDType"/>
    <xsd:element name="CompanyID" type="prov:CompanyIDType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="UserNameType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:complexType name="UserNamePasswordType">
  <xsd:complexContent>
    <xsd:restriction base="prov:AuthenticationType">
      <xsd:sequence>
        <xsd:element name="Role" type="prov:RoleType" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="UserName" type="prov:UserNameType"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="AddressLocationType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="work"/>
    <xsd:enumeration value="mobile"/>
    <xsd:enumeration value="personal"/>
    <xsd:enumeration value="billto"/>
    <xsd:enumeration value="shipto"/>
    <xsd:enumeration value="saletto"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="ITMLFaultDetailType">
  <xsd:sequence>
    <xsd:element name="faultcode" type="prov:faultcodeType"/>
    <xsd:element name="faultstring" type="xsd:string"/>
    <xsd:any namespace="##any" processContents="lax"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="faultcodeType">
  <xsd:restriction base="xsd:QName">
    <xsd:enumeration value="InvalidUserID"/>
    <xsd:enumeration value="InvalidUserInfo"/>
    <xsd:enumeration value="DuplicateUserID"/>
    <xsd:enumeration value="UserNotDisabled"/>
  </xsd:restriction>

```

```

    <xsd:enumeration value="UserNotActive"/>
    <xsd:enumeration value="InvalidCompanyID"/>
    <xsd:enumeration value="DuplicateCompanyID"/>
    <xsd:enumeration value="CompanyNotDisabled"/>
    <xsd:enumeration value="CompanyNotActive"/>
    <xsd:enumeration value="InvalidServiceID"/>
    <xsd:enumeration value="InvalidNewService"/>
    <xsd:enumeration value="ServiceNotCreated"/>
    <xsd:enumeration value="ServiceNotActive"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="txidType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[a-z]{3}:[0-9]{2}:[0-9]{2}:[0-9]{2}:[0-9]{2}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="US-PostalCodeType">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:pattern value="[0-9]{5}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="NA-POBoxType">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:pattern value="[0-9]{5}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="NA-PhoneNumberType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\([0-9]{3}\)[0-9]{3}-[0-9]{4}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="US-SSNType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]{3}-[0-9]{3}-[0-9]{4}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="US-StateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="AL"/>
    <xsd:enumeration value="AK"/>
    <xsd:enumeration value="AZ"/>
    <xsd:enumeration value="AR"/>
    <xsd:enumeration value="CA"/>
    <xsd:enumeration value="CO"/>
    <xsd:enumeration value="CT"/>
    <xsd:enumeration value="DE"/>
    <xsd:enumeration value="DC"/>
    <xsd:enumeration value="FL"/>
    <xsd:enumeration value="GA"/>
    <xsd:enumeration value="HI"/>
    <xsd:enumeration value="ID"/>
    <xsd:enumeration value="IA"/>
    <xsd:enumeration value="IL"/>
    <xsd:enumeration value="IN"/>
    <xsd:enumeration value="KS"/>
    <xsd:enumeration value="KY"/>
    <xsd:enumeration value="LA"/>
    <xsd:enumeration value="MA"/>
    <xsd:enumeration value="ME"/>
    <xsd:enumeration value="MD"/>
    <xsd:enumeration value="MI"/>
    <xsd:enumeration value="MN"/>
    <xsd:enumeration value="MO"/>
    <xsd:enumeration value="MT"/>
    <xsd:enumeration value="MS"/>
    <xsd:enumeration value="NC"/>
  </xsd:restriction>

```

```

<xsd:enumeration value="ND"/>
<xsd:enumeration value="NH"/>
<xsd:enumeration value="NJ"/>
<xsd:enumeration value="NM"/>
<xsd:enumeration value="NB"/>
<xsd:enumeration value="NV"/>
<xsd:enumeration value="NY"/>
<xsd:enumeration value="OH"/>
<xsd:enumeration value="OK"/>
<xsd:enumeration value="OR"/>
<xsd:enumeration value="RI"/>
<xsd:enumeration value="PA"/>
<xsd:enumeration value="SC"/>
<xsd:enumeration value="SD"/>
<xsd:enumeration value="TN"/>
<xsd:enumeration value="TX"/>
<xsd:enumeration value="UT"/>
<xsd:enumeration value="VT"/>
<xsd:enumeration value="VA"/>
<xsd:enumeration value="WA"/>
<xsd:enumeration value="WV"/>
<xsd:enumeration value="WI"/>
<xsd:enumeration value="WY"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="OrganizationInfoType">
  <xsd:sequence>
    <xsd:element name="Department" type="xsd:string"/>
    <xsd:element name="EmployeeNumber" type="xsd:string"/>
    <xsd:element name="EmploymentCategory" type="xsd:string"/>
    <xsd:element name="Manager" type="prov:UserIDType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="OrganizationDatesType">
  <xsd:sequence>
    <xsd:element name="StartActiveDate" type="xsd:string"/>
    <xsd:element name="HireDate" type="xsd:string" minOccurs="0"/>
    <xsd:element name="EndActiveDate" type="xsd:string" minOccurs="0"/>
    <!-- No HireDate means StartActiveDate should be used-->
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

---

## ITMLAddCompany.xml Sample Document

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.0 NT (http://www.xmlspy.com) by David Orchard (Jamcracker) -->
<!-- Instance of a fully formed add Company request -->
<prov:addCompany xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns="http://www.itml.org/ns/provisioning/" xsi:noNamespaceSchemaLocation="ITMLProvMethods.xsd">
  <prov:Company>
    <prov:CompanyID>JC</prov:CompanyID>
    <prov:Name>Jamcracker</prov:Name>
    <prov:CompanyAddressList>
      <prov:FixedAddress>
        <prov:AddressLocation>billing</prov:AddressLocation>
        <prov:CareOfName>Jamcracker Inc</prov:CareOfName>
        <prov:AddressLine1>935 Stewart Drive</prov:AddressLine1>
        <prov:AddressLine2/>
        <prov:City>Sunnyvale</prov:City>
        <!-- Could be Province if Canada -->
        <prov:Country>USA</prov:Country>
        <prov:Email>dorchard@jamcracker.com</prov:Email>
        <prov:State>CA</prov:State>
        <prov:PostalCode>94086</prov:PostalCode>
        <prov:Phone>(123)333-3333</prov:Phone>
        <prov:Fax>(123)333-4444</prov:Fax>
      </prov:FixedAddress>
      <prov:FixedAddress>
        <prov:AddressLocation>shipping</prov:AddressLocation>
        <prov:CareOfName>Jamcracker Inc</prov:CareOfName>
        <prov:AddressLine1>935 Stewart Drive</prov:AddressLine1>
        <prov:AddressLine2/>
        <prov:City>Sunnyvale</prov:City>
        <!-- Could be Province if Canada -->
        <prov:Country>USA</prov:Country>
        <prov:Email>dorchard@jamcracker.com</prov:Email>
        <prov:State>CA</prov:State>
        <prov:PostalCode>94086</prov:PostalCode>
        <prov:Phone>(123)333-3333</prov:Phone>
        <prov:Fax>(123)333-4444</prov:Fax>
      </prov:FixedAddress>
    </prov:CompanyAddressList>
  </prov:Company>
</prov:addCompany>
```

---

## ITMLAddCompanyResponse.xml Sample Document

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ITML Provisioning addCompany Response Sample -->
<prov:addCompanyResponse xmlns="http://www.itml.org/ns/provisioning"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ITMLProvMethods.xsd">
  <prov:CompanyID IsDefinedByPartner="true">JCXYZ</prov:CompanyID>
</prov:addCompanyResponse>
```



---

## Conformance

In addition to the conformance constraints specified in this document and in normative references, the following conformance tests are part of this specification.

Jamcracker is required to send fully populated content messages. Partners are required to understand fully populated content messages. There is no customization of the content messages permitted.

The following script is a minimum conformance test:

1. AddCompany
2. DeleteCompany
3. AddCompany
4. AddCompany
  - a. Duplicate to previous company
  - b. Result should be error
5. AddUser
6. DeleteUser
7. AddUser
8. AddUser
  - a. Duplicate to previous User
  - b. Result should be error
9. Disable User
  - a. Attempt to access services
  - b. Result should be error
10. Enable User
  - a. Attempt to access services
  - b. Result should be successful
11. Modify User
  - a. Change User Information
12. Add Service for Company
13. Activate Service for Company
14. Upgrade/Downgrade Service for Company
15. Suspend Service for Company
16. Enable Service for Company
17. Delete Service for Company
18. Order a Service for User
19. Upgrade/Downgrade a Service for User
20. Suspend a Service for User
21. Enable a Service for User
22. Delete a Service for User

---

# Appendix

## Additional Normative Documents

The following documents are considered part of this specification, but are not bundled with it in the interests of space constraints. It is suggested that interested parties should print the documents only when necessary:

Additional documents:

- ITMLCRMMethods.xsd
- ITMLDiCartaProvMethods.xsd
- ITMLManageMarkProvMethods.xsd
- ITMLAddCRMAccount.xml
- ITMLAddCRMContact.xml
- ITMLAddUser.xml
- ITMLAddUserResponse.xml
- ITMLAddUserResponseError.xml
- ITMLDeleteUser.xml
- ITMLDiCarataAddCompanyResponse.xml
- ITMLManageMarkAddUser.xml
- ITMLUpdateCompany-Disable.xml
- ITMLUpdateCompany-AddService.xml
- ITMLUpdateUser-AddService.xml
- ITMLUpdateUser-ModifyService.xml
- ITMLUpdateUser-Disable.xml
- ITMLUpdateUserResponse-Error.xml

## Issues

1. Effective Dating
2. Optional commands
3. Namespace uniqueness guarantees
4. Name Mangling with alternative login
5. Send Passwords
6. Multiple usernames per user
7. Authentication protocol – EmployEase would prefer username/password in HTTP Header
- 8.

## Typical information for each Partner (non-normative)

The following is a list of information that is typically needed in a Jamcracker/partner communication:

- Protocol – fixed at HTTPS currently
- Who assigns username
- Who assigns companyID
- Partner URL to send Provisioning messages to
- Jamcracker username/password
- Partner Test URL
- Partner Test username/password
- Interaction timeout

## **Futures (non-normative)**

Provisioning will be extended in the future. There are many different alternatives to follow. This section gives readers some idea of the possibilities.

- Addition and modification of Service information.
- A formal refinement mechanism, allow Partners to specify a subset of the data elements so they don't get more data than they need, and users don't have to enter extra data.
- transform the data before it reaches partners, ie combining first name and last name into a single string.
- Bulk loading
- Read information required
- User information modified at Partner then sent to Jamcracker
- Send User passwords

## **Partner Guidelines (non-normative)**

The application provider should consider hiding / disabling the ability to add/delete/enable/disable and modify company's and users within accounts that were created by Jamcracker.

The application provider should disable logons for users within accounts that were created by Jamcracker.

## **Credits**

This specification has been greatly helped by the following people and affiliations:

Vinay Singla - Jamcracker  
Viswanath Reddy – Jamcracker  
Camie Hackson – ManageMark  
Chris Haddad – EmployEase  
Jon Finegold – OpenAir

## **Revision History**

2000:

Aug 2 – Initial Version  
Aug 3 to 22 – Many revisions

Sept 21-Oct 6: Added Issues Appendix with titles but not links. Updated Schemas: 1) Made update content model a choice 2) Errors in responses. Added UpdateUser, UpdateUserResponse.xml, DeleteUser.xml, UpdateCompany samples. Added partner guidelines appendix. Added schema information section with subsections on effective dating, users/roles/credentials.

Oct 10: Removed passwords from schemas and added issue, Added partner defined roles, Added issue about multiple usernames.

Oct22-Dec 5: Updated for Version 0.8. Added information model, data dictionary, service subscriptions, crm information. Info model changes: split users into actors with roles, separated users from actors,

Dec 20: Added many fields – departments, phone extension, pager, manager, employee #, city, category, hire date. Added multiple roles to authentication.

Feb 14-March 2nd 2001: Updated to XML Schema CR version, changed asp specific data collection to schema wildcard. Added design goals. Removed effective dating.