UH Development   
Webservices for CRM integration  
for One Housing Group

Dec 1st 2011



# UH Development: CRM Webservices Development

History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date Issued** | **Author** | **Reason** | **Sections affected** |
| 1 | 1/12/2011 | Jeremy Riches | First draft of document |  |
| 1.1 | To 5/1/12 | Jeremy Riches | Updates following internal review |  |
| 1.2 | 12/1/12 | Jeremy Riches | Rename doc and split out works order reqs |  |
| 1.3 & 1.4 | 17/1/12 | Jeremy Riches | Updates following development review |  |
|  |  |  |  |  |

# Summary

This document details the development requirement for a set of new webservices to be added into Universal Housing, for the purpose of integrating with CRM systems. The development will be focussed around the requirements for the “One Housing Group”, but will also include some additional services and requirements to provide the basis of a generic solution.

The first part of this document describes at a high level an overview of the services that will be included in the package, with more detailed information relating to each service in subsequent sections.

The contents of this document are based upon a requirements document produced by Mark Beanland of One Housing Group dated 28th October 2011; subsequent analysis by Civica; and discussions that have taken place up to the date of issue of this version of the document, as detailed in the document history. The solution outlined in this document has been designed to fulfil the requirements for One Housing Group, but re-worked into a generic collection of webservices.

\*\*This document details the requirements for the first phase of development which relate to Contact and Property related services. A subsequent document will provide details relating to the works order services.

# Webservice Integration Approach

The UH webservices will be developed in C#.NET 4.0 using WCF to be hosted on an IIS Server. .NET Framework v4 or greater will need to be installed on the server prior to the web service installation.

Civica will use a framework within UH (currently in the process of development) for managing and controlling the use of these webservices, along with a wider set of webservices that will be developed into the UHT application. This framework is summarised in section 6. This framework includes application level authentication (described below) for the webservices, maintenance of associated parameters, managing which webservices are enabled, and an Audit mechanism for optionally logging calls made to the webservices. A new console screen within UHT will manage all of these elements.

From an authentication and security perspective, the following will be able to be setup via the new console:

1. A master password can optionally be “switched on” for use. If so, then this password must be passed in to each webservice call
2. A number of “webservice users” can optionally be used. These are setup in the new UH console, where they can also be linked to a UH user record. If this functionality is used, then a valid webservice user name will need to be passed in to each webservice call.
3. To go alongside (b), a user password can optionally be setup and maintained. If this is used, then a valid password will need to be passed in the webservice with the webservice user name.
4. If webservice user name functionality is used, then in the authentication services package, there is the ability to “log on” and “log off”.... this validates the username & password. Optionally this can also return a unique session id value (via the log on process). This session token can then be used as an additional security measure, to pass in to each webservice call. The session token will remain active for the duration of the session (until log off) or if required an “inactive” period has elapsed after which the session token will be cleared and any subsequent calls to the webservices will result in a session token error.

Where services are being exposed externally, we would recommend securing at the transport level eg by SSL.

All webservices include the option to pass in an identifier for the source system making the call, along with any key data reference.... this information will be stored in the Audit records (if auditing is turned on) for traceability.

# Overview of Webservices Provided

The services highlighted in green below will be included in the development work for One Housing Group. The additional services listed may be developed by Civica at some future date to extend the services for additional/generic use. Civica make no guarantee that these additional services will be developed at a future date and make no commitments as to the commercial arrangement under which they may be available/provided. They are simply referred to here for indicative purposes only.

## Authentication Services A set of standard services within the Civica UH webservices framework for authenticating and controlling use of the webservices.

## Auditing Services

A set of standard services within the Civica UH webservices framework for writing and reading records for usage of webservices.

## Contact Communications Services – PHASE ONE DEVELOPMENT *(Only those in green are included in this development)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **O.H.G. method** | **Civica Method Name** | **Description** |
| **Read Services** | | | |
| CC01 | <not requested> | GetCommsEntryDetail | Returns comms entry detail record |
| CC02 | <not requested> | GetCommsEntryEntityList | Returns list of comms entry records per entity (eg household) |
| **Write Services** | | | |
| CC51 | createContactDetail | CreateCommsEntry | Creates a new record in the communication entries table |
| CC52 | updateContactDetail | UpdateCommsEntry | Updates an existing record in the communication entries table |
| CC53 | deleteContactDetail | DeleteCommsEntry | Deletes an existing record in the communication entries table |

## Household Services – PHASE ONE DEVELOPMENT *(Only those in green are included in this development)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **O.H.G. method** | **Civica Method Name** | **Description** |
| **Read Services** | | | |
| HH01 | <not requested> | GetHouseholdDetail | Returns household detail info |
| HH02 | <not requested> | GetHouseholdMemberDetail | Returns Member detail info |
| HH03 | <not requested> | GetHousholdMemberList | Returns list of Members at Household |
| **Write Services** | | | |
| HH51 | <not requested> | CreateHousehold | Creates a new record in the Household table + initial household members |
| HH52 | updatePersonInfo (in part) | UpdateHousehold | Updates an existing record in the Household members table |
| HH53 | <not requested> | DeleteHousehold | Deletes an existing record in the Household members table |
| HH61 | <not requested> | CreateHouseholdMember | Creates a new record in the Household Member table |
| HH62 | updatePersonInfo (in part) | UpdateHouseholdMember | Updates an existing record in the Household Member table |
| HH63 | <not requested> | DeleteHouseholdMember | Deletes an existing record in the Household Member table |

## Property Services – PHASE ONE DEVELOPMENT *(Only those in green are included in this development)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **O.H.G. method** | **Civica Method Name** | **Description** |
| **Read Services** | | | |
| PM01 | <not requested> | GetPropertyDetails | Returns property details |
| PM02 | <not requested> | GetPropertyListBySearch | Returns list of properties per search criteria |
| PR01 | <not requested> | GetRentDetails | Returns rent details for property |
| **Write Services** | | | |
| PM51 | <not requested> | CreateProperty | Create a new property record + Associated Rent Record |
| PM52 | UpdatePropertyInfo | UpdateProperty | Update an existing property record |
| PM53 | <not requested> | DeleteProperty | Delete an existing property & associated rent record |
| PR52 | UpdatePropertyInfo | UpdateRentDetails | Update an existing rent record |

## Repairs Services – PHASE TWO DEVELOPMENT

Covered in a separate specification document

# Naming Conventions

As a default we recommend services will be published with an address format of:

[http://[server:port]/[environment]-civica-uh-gateway/webservices](http://[server:port]/%5benvironment%5d-civica-uh-gateway/webservices)

Each webservice will follow a naming convention of :

* Create<Entity>
* Update<Entity>
* Delete<Entity>
* Get<Entity>

Data Transfer Objects:

Generally speaking the various Dto objects will be designed such that properties that are references to other objects \ tables will end with “Reference”. Properties to store lookup codes will end with “Code”.

E.g. on the PropertyDto the property \ element to store the unique ref of the household object \ table (field name house\_ref) would be HouseholdReference. A lookup code such as rep\_officer would be RepairOfficerCode.

# Webservice Response

The webservice will respond to advise whether the transaction completed successfully. A bool value will be returned to indicate success or failure along with a SOAP fault (SOAP-ENV fault tag) with faultcode and faultstring to be defined around the errors defined throughout the spec below...

Error Numbers:

Errors listed with a value of 900 or above are errors which may apply across all webservices. Error 999 is a general non-specific exception error.

Errors pertaining to each individual webservice (eg validation type errors) are to be sequentially numbered from 001 for each service.... Note: error 001 on webservice A may therefore have a different meaning to error 001 on webservice B.

# UH Webservices framework

## The webservices are to be developed alongside the UH webservice framework (not detailed in this document).

In summary this framework provides standard associated functionality for:

* Controlling and licensing use of the webservices
* Managing authentication via master password, user names & passwords, and tokens.
* Auditing into a log table
* Providing associated parameters.

An admin console screen will be developed in UHT to manage the above, and released in version 2012.2 of UHT.

Our understanding for OHG is that only a master password will be used (which will not be encrypted) – the services will be deployed across a OHG LAN and not exposed externally.

# Authenticating the use of each webservice

Each webservice will call a standard procedure to validate that the service can run. Validation checks that the service is licensed and enabled, and also checks any authentication that is switched on.

# Detailed Webservice Requirements

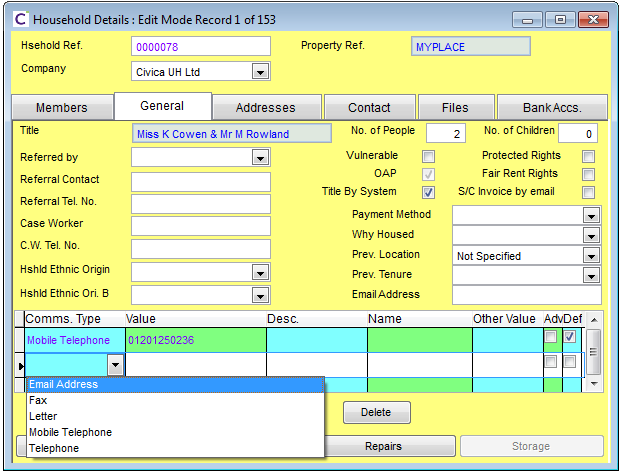
## CC51: CreateCommsEntry

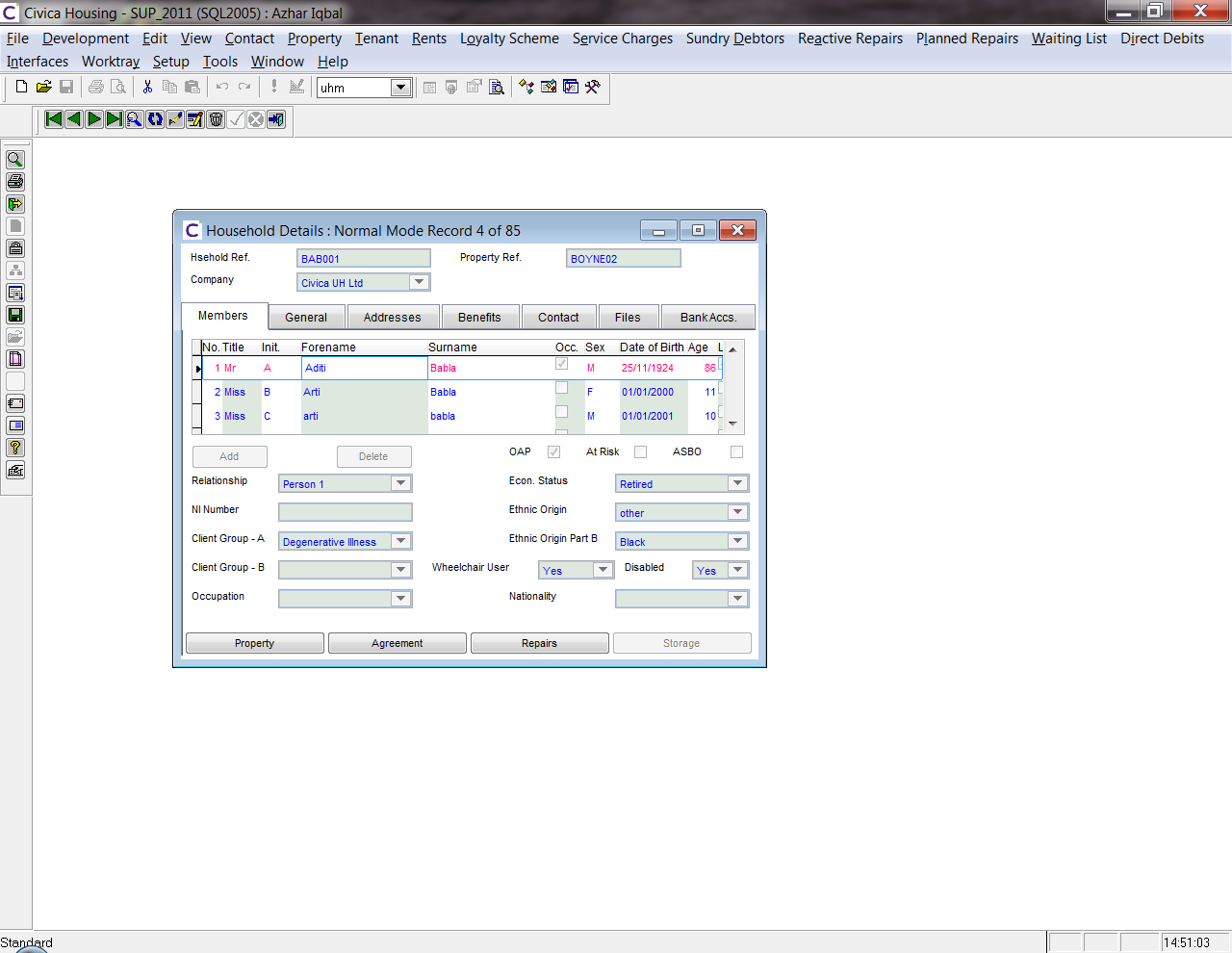
Purpose/Requirement for O.H.G:

To add a communications entry into UH – either against the household or against a specific person.

Proposed Solution:

An example screen shot of the related UHT screen is below:





Note: In UHT, **communications cannot be associated directly with an individual person** – they are linked to **groups** of people or other entities. Currently communications entries can be added to :

* Households,
* Waiting List Applicants
* Properties
* Suppliers (for Repairs)
* Homelessness

The webservice will therefore accept a (mandatory) value to determine which entity the communications entry is required for eg “Household”. All communications records will therefore be against the entity (eg Household). The key reference of that entity will also be required.

One Housing Group will utilise a redundant field (eg Description??) to populate a person reference to facilitate a link to a specific person. (No additional development required for this, as OHG will simply pass in the value).

Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Data Object: CommsEntryDto (see Appendix A)
    - Contact Key Type (Household, Waiting List, Homeless, Supplier, Property)
    - Contact Key Ref (eg Household Ref)
    - Name (Person Name)
    - Communication Type (Email, Fax, Letter....)
    - Communication Value (eg 0123 232322)
    - Description (free entry text)
    - Other Value
    - Advocate (Indicator for person acting on someone’s behalf eg solicitor or a relative)
    - Default (Indicator for the default communication)
    - Userfields (list of user fields):
      * Fieldname
      * Value
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)
  + Comms Entry Sid (Unique id for record)

Processing

* Records are stored in the UHT Comms table.
* Contact Key Type is mandatory and must be a recognised entity, else webservice will error
* Contact Key Reference is mandatory and must exist, else the webservice will error
* The entity\_sid of the new comms entry will be set to the sid of the related entity. E.g. in the case of a household it will be set to househ.househ\_sid.
* The table below details the processing based on the Name field passed in, along with a new system parameter as to whether the passed in Name must exist in the list of contact names against the entity group:

|  |  |  |
| --- | --- | --- |
| **Name** | **Sys Param:  Name must exist** | **Outcome for creation of comms entry record** |
| Null | Yes | Error returned – Name is Required |
| Null | No | Populate comms record with null Name |
| Passed In | Yes | Validate Name exists in person (member) list for entity (eg household) and error if it doesnt exist. |
| Passed In | No | Populate comms record with passed in Name |

* Default – if Boolean:True is passed in, then this is the default contact entry, and any previous record for the entity (eg household) with the default flag set needs to have its flag reset to false.

Errors

* 1 : CommsEntry.MissingSource : Missing Source System Identifier
* 10 : CommsEntry.InvalidKeyEntity : Unknown Contact Key Entity
* 20: CommsEntry.NameNotFound: Name not found in List of Members
* 30: CommsEntry.InvalidCommsType: Unknown Communications Type
* 31: CommsEntry.MissingDescription: No description provided
* 32: CommsEntry.InvalidDefaultFlag: Default Flag must be True or False
* 40: CommsEntry.InvalidUserField: User Field not recognised
* 999: CommsEntry.Unknown : Unknown processing error

## CC52: UpdateCommsEntry

Webservice will be similar to CC51, except for updating an existing record.   
  
The “Comms Entry Sid” will determine the record to update. The calling system will either need to store the Sid reference in order to update the record, or alternatively use one of the “get....” services to retrieve the Sid which needs to be updated.

Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Data Object: CommsEntryDto
    - Comms Entry Sid (UH unique record id)
    - *Not Updateable: Contact Key Type*
    - *Not Updateable: Contact Key Ref*
    - Name (Person Name)
    - Communication Type (Email, Fax, Letter....)
    - Communication Value (eg 0123 232322)
    - Description (free entry text)
    - Other Value
    - Advocate (Indicator for contact acting on behalf of a member)
    - Default (Indicator for the default communication)
    - Userfields (list of user fields):
      * Fieldname
      * Value
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

\*\*see logic for service CC51

Errors

* 1 : CommsEntry.MissingSource : Missing Source System Identifier
* 5 : CommsEntry.InvalidSid : Unknown Comms Entry ID
* 20: CommsEntry.NameNotFound: Name not found in List of Members
* 30: CommsEntry.InvalidCommsType: Unknown Communications Type
* 31: CommsEntry.MissingDescription: No description provided
* 32: CommsEntry.InvalidDefaultFlag: Default Flag must be True or False
* 40: CommsEntry.InvalidUserField: User Field not recognised
* 999: CommsEntry.Unknown : Unknown processing error

## CC53: DeleteCommsEntry

Webservice will be similar to CC52, except for deleting an existing record.   
  
The “Comms Entry Sid” will determine the record to update. The calling system will either need to store the Sid reference in order to update the record, or alternatively use one of the “get....” services to retrieve the Sid which needs to be updated.

Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Comms Entry Sid (UH unique record id)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Locate the UH Comms record relating to the Comms Entry Sid, and if found delete record, else error.

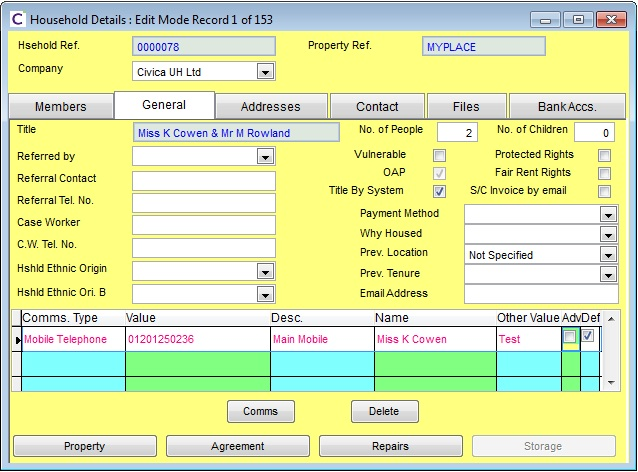
\*\*Deletion must always create an audit record into webservice\_audit.

Errors

* 1 : CommsEntry.MissingSource : Missing Source System Identifier
* 5 : CommsEntry.InvalidSid : Unknown Comms Entry ID
* 999: CommsEntry.Unknown : Unknown processing error

## HH51: CreateHousehold

*<not required for O.H.G>*

This service is to facilitate the creation of a Household record, and associated Member records (at least one member record is required).  
  


Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Data Object: HouseholdDto
    - (see Appendix A)
  + Data Object: HouseholdMemberDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)
  + Household Ref

Processing

Creation of record in UHT table: househ

Errors

TBA

## HH52: UpdateHousehold

Relates to O.H.G..... also see CH62

Purpose/Requirement for O.H.G:

This service exists primarily to allow profiling information to be updated against household or household member.

Proposed Solution:

*This relates in part to O.H.G. requirement service “updatePersonInfo”. This requirement has been split into two separate webservices – one to deal with updating Household info (this service) – and an additional service to update the Household Member (see service CH62).*   
  
The O.H.G requirements suggested a collection of fieldname & value pairs.... Civica propose that the service should specifically reference an object (dto – data transfer object) containing the standard household fields with user defined fields (“u\_” fields) handled through the fieldname and value pairs – this approach is more generic; enables the standard fields to be easily referenceable; and Civica can develop business logic/validation around the standard fields.

The update will utilise “house\_ref” as the key to identify the record to be updated. The webservice will pass the associated HouseholdDto object via the webservice.

Note: unlike the “CreateHousehold” webservice (CH51) where additionally household member records can be passed in, for this update service only the Household header record can be updated. A separate service exists for updating Household members.

Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Data Object: HouseholdDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Locate the related househ record via the house\_ref key value passed in, and update the record based on the values in the dto  
  
Errors  
  
TBA

## HH53: DeleteHousehold

*<not required for O.H.G>*

This service is to facilitate the deletion of a Household record. As this service is not required by O.H.G, this will be specified at a future date.

## HH61: CreateHouseholdMember

This webservice is to enable additional members to be added to a household.

<not required for O.H.G.>

## HH62: updateHouseholdMember

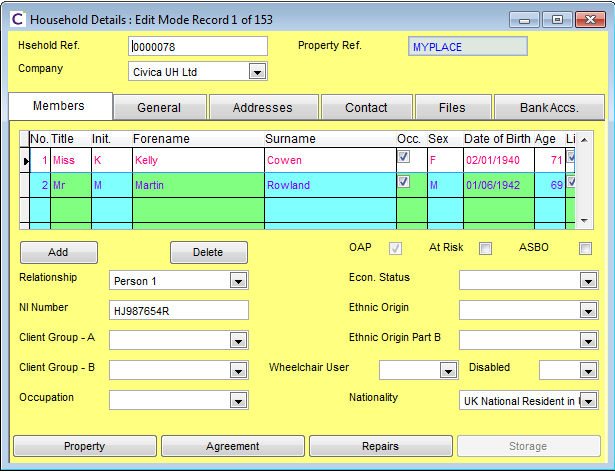
Relates to O.H.G..... also see CH62

Purpose/Requirement for O.H.G:

This service exists primarily to allow profiling information to be updated against household or household member.

Proposed Solution:

*This relates in part to O.H.G. requirement service “updatePersonInfo”. This requirement has been split into two separate webservices – one to deal with updating Household info (see service CH52) – and an additional service to update the Household Member (this service).*   
  
The O.H.G requirements suggested a collection of fieldname & value pairs.... Civica propose that the service should specifically reference an object (dto – data transfer object) containing the standard household member fields with user defined fields (“u\_” fields) handled through the fieldname and value pairs – this approach is more generic and enables the standard fields to be easily referenceable.   
  
The update will utilise “house\_ref” + “person no” as the combined key to identify the record to be updated. The webservice will pass the associated HouseholdMemberDto object via the webservice.



Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Data Object: HouseholdMemberDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Locate record in Member table based on Household reference + person no, and update the record based on the values in the dto  
  
Errors  
  
TBA

## HH63: DeleteHouseholdMember

This webservice is to enable additional members to be added to a household.

<not required for O.H.G.>

## PM51: CreateProperty

This webservice is to facilitate the creation of Property records. If the property is lettable, then a rent table record will be created. A new Void Monitoring record will also need to be created.

<not required for O.H.G.>

## PM52: UpdateProperty

Purpose/Requirement for O.H.G:

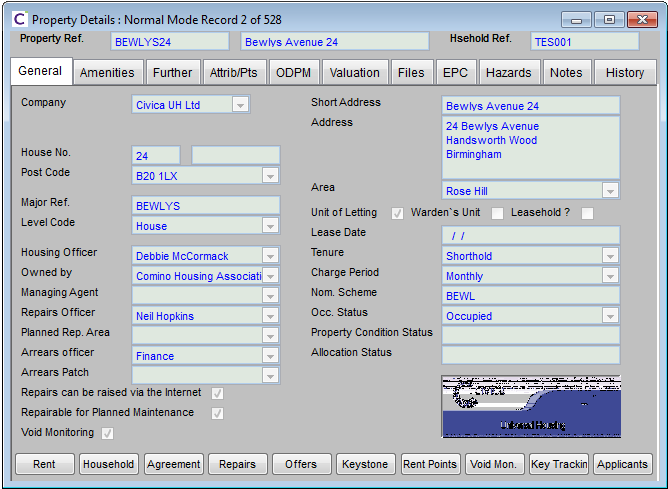
This service exists primarily to allow profiling information to be updated against a property.

Proposed Solution:

The O.H.G requirements suggested a collection of fieldname & value pairs.... Civica propose that the service should specifically reference an object (dto – data transfer object) containing the standard property fields with user defined fields (“u\_” fields) handled through the fieldname and value pairs – this approach is more generic; enables the standard fields to be easily referenceable; and Civica can develop business logic/validation around the standard fields.

The update will utilise “prop\_ref” as the key to identify the record to be updated. The webservice will pass the associated PropertyDto object via the webservice.

Note: UHT stores rent related information in an associated table, linked by “prop\_ref”. Service PR52 provides a means of updating data into this related table.



Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Data Object: PropertyDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Locate the related property record via the prop\_ref key value passed in, and update the record based on the values in the dto  
  
Errors  
  
TBA

## PM53: DeleteProperty

This webservice is to facilitate the deletion of Property records. Records can only be deleted if the property has at no point been associated with a Tenancy and no void loss has been accrued against it.

If deletion is permitted then the associated rent record should also be deleted

<not required for O.H.G.>

## PR52: UpdateRentInfo

*Note: there is no requirement for a “CreateRentInfo” service, as the rent record is to be created as part of the “CreateProperty” service or as part of the process of making a property lettable.*

This webservice enables rent information associated with a property to be updated.

Proposed Solution:

The update will utilise “prop\_ref” as the key to identify the record to be updated into the rent table. The webservice will pass the associated PropertyRentDto object via the webservice.

Webservice

* Data Input Requirements
  + Authentication Elements
    - Master Webservices Password
    - Webservice User ID
    - User Session Token
  + Source System Reference
    - External Source System
    - External System Unique Record ID / Key
  + Data Object: PropertyRentDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Locate the related rent record via the prop\_ref key value passed in, and update the record based on the values in the dto  
  
Errors  
  
TBA

# Base Data Web Services

A number of webservices to provide base data (eg codes, types etc) may be developed by Civica at a future date to extend the package as part of a generic solution.  
  
These services are not required by O.H.G. and will not be provided under this current development but some are summarised below. The list will be extended during future design and the services will be developed at a future date.

* Comms Types
* Payment Methods
* Ethnic Origins
* Tenure Types
* Housing Officers
* Charge Periods
* Occupation Status
* Visit Types

etc

# Appendix A – DTO Object Structures

## CommsEntryDto

<CommsEntryDto>

<CommsEntrySid> </CommsEntrySid>

<ContactKeyType> </ContactKeyType>

<ContactKeyRef> </ContactKeyRef>

<Name> </Name>

<CommType> </CommType>

<CommValue> </CommValue>

<Description> <Description>

<OtherValue> </OtherValue>

<Advocate> </Advocate>

<Default> </Default>

<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

</CommsEntryDto>

*Note: CommsEntrySid (unique UH record reference) will be passed in as Null through this service as this value is assigned by UH – this value will be used for locating the record for updates/deletion.*

## HouseholdDto

<HouseholdDto>

<HouseholdReference> </HousholdReference>

<Description> </Description>

<ReferralContactName> </ReferralContactName>

<ReferralContactTelNo> </ReferralContactTelNo>

<CaseWorkerContactName> </CaseWorkerContactName>

<CaseWorkerContactTelNo> </CaseWorkerContactTelNo>

<IsFairRights> </IsFairRights>

<IsProtectedRights> </IsProtectedRights>

<IsDescSysGenerated> </IsDescSysGenerated>

<IsVulnerable> </IsVulnerable>

<IsSCInvoiceEmailed> </IsSCInvoiceEmailed>

<Size> </Size>

<NumberOfChildren> </NumberOfChildren>

<ContactReference> </ContactReference>

<PropertyReference> </PropertyReference>

<Household Sid> </Household Sid>

<CompanyCode> </CompanyCode>

<DisplayCompanyCode> </DisplayCompanyCode>

<CorrespondenceAddress> </CorrespondenceAddress>

<CorrespondencePostCodeValue> </CorrespondencePostCodeValue>

<CorrespondenceDesignation> </CorrespondenceDesignation>

<CorrespondencePreamble> </CorrespondencePreamble>

<ForwardingAddress> </ForwardingAddress>

<ForwardingPostCodeValue> </ForwardingPostCodeValue>

<ForwardingDesignation> </ForwardingDesignation>

<ForwardingPreamble> </ForwardingPreamble>

<NextOfKinAddress> </NextOfKinAddress>

<NextOfKinPostCodeValue> </NextOfKinPostCodeValue>

<NextOfKinDesignation> </NextOfKinDesignation>

<NextOfKinPreamble> </NextOfKinPreamble>

<NextOfKinName> </NextOfKinName>

<NextOfKinPhone> </NextOfKinPhone>

<PropertyAccessDetails> </PropertyAccessDetails>

<PropertyAccessMondayAM> </PropertyAccessMondayAM>

<PropertyAccessMondayPM> </PropertyAccessMondayPM>

<PropertyAccessMondaySR> </PropertyAccessMondaySR>

<PropertyAccessTuesdayAM> </PropertyAccessTuesdayAM>

<PropertyAccessTuesdayPM> </PropertyAccessTuesdayPM>

<PropertyAccessTuesdaySR> </PropertyAccessTuesdaySR>

<PropertyAccessWednesdayAM> </PropertyAccessWednesdayAM>

<PropertyAccessWednesdayPM> </PropertyAccessWednesdayPM>

<PropertyAccessWednesdaySR> </PropertyAccessWednesdaySR>

<PropertyAccessThursdayAM> </PropertyAccessThursdayAM>

<PropertyAccessThursdayPM> </PropertyAccessThursdayPM>

<PropertyAccessThursdaySR> </PropertyAccessThursdaySR>

<PropertyAccessFridayAM> </PropertyAccessFridayAM>

<PropertyAccessFridayPM> </PropertyAccessFridayPM>

<PropertyAccessFridaySR> </PropertyAccessFridaySR>

<PropertyAccessSaturdayAM> </PropertyAccessSaturdayAM>

<PropertyAccessSaturdayPM> </PropertyAccessSaturdayPM>

<PropertyAccessSundayAM> </PropertyAccessSundayAM>

<PropertyAccessSundayPM> </PropertyAccessSundayPM>

<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

</HouseholdDto>

## HouseholdMemberDto

<HouseholdMemberDto>

<HouseholdReference> </HouseholdReference>

<PersonNo> </PersonNo>

<Title> </Title>

<Initials> </Initials>

<Forename> </Forename>

<Surname> </Surname>

<Gender> </Gender>

<NationalInsuranceNo> </NationalInsuranceNo>

<Relationship> </Relationship>

<IsResponsible> </IsResponsible>

<IsLiable> </IsLiable>

<IsOldAgePensioner> </IsOldAgePensioner>

<IsAtRisk> </IsAtRisk>

<IsAsboIssued> </IsAsboIssued>

<IsRegisteredDisabled> </IsRegisteredDisabled>

<DateOfBirth> </DateOfBirth>

<EthnicOriginCode> </EthnicOriginCode>

<EthnicColourCode> </EthnicColourCode>

<OccupationCode> </OccupationCode>

<DisabledQuestionCode> </DisabledQuestionCode>

<WheelChairUserQuestionCode> </WheelChairUserQuestionCode>

<EconomicStatusCode> </EconomicStatusCode>

<NationalityCode> </NationalityCode>

<ClientGroupACode> </ClientGroupACode>

<ClientGroupBCode> </ClientGroupBCode>  
  
<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

</HouseholdMemberDto>

## PropertyDto

<PropertyDto>

<PropertyReference> </PropertyReference>

<RepairSubArea> </RepairSubArea>

<DisplayCompany> </DisplayCompany>

<ManagementScheme> </ManagementScheme>

<PostDesignation> </PostDesignation>

<ShortAddress> </ShortAddress>

<Telephone> </Telephone>

<Condition> </Condition>

<InsurancePolicyNumber> </InsurancePolicyNumber>

<FinanceCompany> </FinanceCompany>

<Subtype> </Subtype>

<EpcCertificateNo> </EpcCertificateNo>

<PracticalCompletionDate> </PracticalCompletionDate>

<HandoverDate> </HandoverDate>

<DecentHomesDateAssessed> </DecentHomesDateAssessed>

<EpcSurveyDate> </EpcSurveyDate>

<EpcRequestDate> </EpcRequestDate>

<EpcRecordedDate> </EpcRecordedDate>

<NherRating> </NherRating>

<CyclicalDue> </CyclicalDue>

<RateValue> </RateValue>

<InsValue > </InsValue >

<Id> </Id>

<Keys> </Keys>

<WalkNo> </WalkNo>

<WalkSequence> </WalkSequence>

<SapRating> </SapRating>

<FeatureNoBedrooms> </FeatureNoBedrooms>

<FeatureNoFloors> </FeatureNoFloors>

<FeatureMinAgeRestriction> </FeatureMinAgeRestriction>

<FeatureMaxAgeRestriction> </FeatureMaxAgeRestriction>

<FeatureFamilySize> </FeatureFamilySize>

<FeatureNoSteps> </FeatureNoSteps>

<DecentHomesYearOfFailure> </DecentHomesYearOfFailure>

<DecentHomesCostNow> </DecentHomesCostNow>

<DecentHomesCostAtYearOfFailure> </DecentHomesCostAtYearOfFailure>

<EpcEnergyRating> </EpcEnergyRating>

<EpcCO2Rating> </EpcCO2Rating>

<IsManagedProperty> </IsManagedProperty>

<IsLetable> </IsLetable>

<IsRtb> </IsRtb>

<IsCoreShared> </IsCoreShared>

<IsLeasehold> </IsLeasehold>

<IsS125> </IsS125>

<IsVoidMonLive> </IsVoidMonLive>

<IsRepairable> </IsRepairable>

<IsNoMaintenanceAllowed> </IsNoMaintenanceAllowed>

<HasAsbestos> </HasAsbestos>

<HasRtbApplication> </HasRtbApplication>

<HasLocalConnection> </HasLocalConnection>

<HasAllocationsPanel> </HasAllocationsPanel>

<CanCreateRepairsOnline> </CanCreateRepairsOnline>

<FeatureHasLounge> </FeatureHasLounge>

<FeatureHasLaundry> </FeatureHasLaundry>

<FeatureHasVisitorBed> </FeatureHasVisitorBed>

<FeatureHasStore> </FeatureHasStore>

<FeatureHasWardenFlat> </FeatureHasWardenFlat>

<FeatureIsSheltered> </FeatureIsSheltered>

<FeatureHasShower> </FeatureHasShower>

<FeatureHasCommLifts> </FeatureHasCommLifts>

<FeatureArePetsAllowed> </FeatureArePetsAllowed>

<FeatureAreChilrenAllowed> </FeatureAreChilrenAllowed>

<FeatureHasGarage> </FeatureHasGarage>

<FeatureHasStairList> </FeatureHasStairList>

<FeatureHasThroughLift> </FeatureHasThroughLift>

<FeatureHasAccessShower> </FeatureHasAccessShower>

<FeatureHasRamp> </FeatureHasRamp>

<FeatureHasHandRails> </FeatureHasHandRails>

<FeatureHasDiningRoom> </FeatureHasDiningRoom>

<FeatureHasKitchenDining> </FeatureHasKitchenDining>

<FeatureHasSecondToilet> </FeatureHasSecondToilet>

<CompanyCode> </CompanyCode>

<ContactKey> </ContactKey>

<HouseholdReference> </HouseholdReference>

<LevelReference> </LevelReference>

<ParentReference> </ParentReference>

<PostCodeValue> </PostCodeValue>

<OwnerReference> </OwnerReference>

<TypeReference> </TypeReference>

<OccupationStatusReference> </OccupationStatusReference>

<WardenReference> </WardenReference>

<LocalAuthorityReference> </LocalAuthorityReference>

<WaterSupplierReference> </WaterSupplierReference>

<BuildingSchemeReference> </BuildingSchemeReference>

<VoidMaintenanceReference> </VoidMaintenanceReference>

<PlannedRepairAreaReference> </PlannedRepairAreaReference>

<ResidentsAssosiationReference> </ResidentsAssosiationReference>

<ArrearsPatchCode> </ArrearsPatchCode>

<ArrearsOfficerCode> </ArrearsOfficerCode>

<RepairOfficerCode> </RepairOfficerCode>

<OwnerConfirmationCode> </OwnerConfirmationCode>

<LettingAreaCode> </LettingAreaCode>

<MaintenanceResponsibilityCode> </MaintenanceResponsibilityCode>

<HeatingCode> </HeatingCode>

<RepairAreaCode> </RepairAreaCode>

<HousingOfficerCode> </HousingOfficerCode>

<AreaOfficeCode> </AreaOfficeCode>

<DecentHomeStatusCode> </DecentHomeStatusCode>  
  
<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

<PropertyDto>

## PropertyRentDto

<PropertyRentDto>

<PropertyReference> </PropertyReference>

<RentgroupReference> </RentgroupReference>

<OccupationStatusReference> </OccupationStatusReference>

<DefaltVoidOccupationStatusReference> </DefaltVoidOccupationStatusReference>

<TenureReference> </TenureReference>

<NewTenureReference> </NewTenureReference>

<HouseholdReference> </HouseholdReference>

<ComanyCode> </ComanyCode>

<ServiceChargePeriodCode> </ServiceChargePeriodCode>

<DateOccupationStatusFromDate> </DateOccupationStatusFromDate>

<MajorPhaseFromDate> </MajorPhaseFromDate>

<LastReviewDate> </LastReviewDate>

<NextReviewDate> </NextReviewDate>

<NewTenureEffectiveDate> </NewTenureEffectiveDate>

<ValuationDate> </ValuationDate>

<CblLetDate> </CblLetDate>

<ServiceChargeLastReviewDate> </ServiceChargeLastReviewDate>

<ServiceChargeNextReviewDate> </ServiceChargeNextReviewDate>

<RentRegistrationDateFirstRegistered> </RentRegistrationDateFirstRegistered>

<RentRegistrationDate> </RentRegistrationDate>

<RentRegistrationDueDate> </RentRegistrationDueDate>

<RentRegistrationApplicationDate> </RentRegistrationApplicationDate>

<AccomodationTypeCode> </AccomodationTypeCode>

<RentSettingMethodCode> </RentSettingMethodCode>

<BuildTypeCode> </BuildTypeCode>

<FloorLevelCode> </FloorLevelCode>

<DesignTypeCode> </DesignTypeCode>

<BarrierFreeCode> </BarrierFreeCode>

<ServiceChargeMethodCode> </ServiceChargeMethodCode>

<ServiceChargeBalMethodCode> </ServiceChargeBalMethodCode>

<AllowFreePeriods> </AllowFreePeriods>

<IsMobilityUnit> </IsMobilityUnit>

<IsWheelchairAdapted> </IsWheelchairAdapted>

<IsFurnished> </IsFurnished>

<IsLocalLetting> </IsLocalLetting>

<IsRtbEligible> </IsRtbEligible>

<IsSuspendedFromCbl> </IsSuspendedFromCbl>

<ValuationIsEstimated> </ValuationIsEstimated>

<ValuationIsActual> </ValuationIsActual>

<Notes> </Notes>

<ShortAddress> </ShortAddress>

<DisplayCompany> </DisplayCompany>

<RR1RoomDetails> </RR1RoomDetails>

<RR1OtherDetails> </RR1OtherDetails>

<RR1LandlordSharedAccom> </RR1LandlordSharedAccom>

<RR1OtherSharedAccom> </RR1OtherSharedAccom>

<RR1CouncilTaxToLandlord> </RR1CouncilTaxToLandlord>

<RR1CouncilTaxFromLandlord> </RR1CouncilTaxFromLandlord>

<RR1FurnitureDetails> </RR1FurnitureDetails>

<RR1TenancyStartDate> </RR1TenancyStartDate>

<RR1LandlordRepairs> </RR1LandlordRepairs>

<RR1TenantRepairs> </RR1TenantRepairs>

<RR1RatesDetails> </RR1RatesDetails>

<RR1ReRegisteringDetails> </RR1ReRegisteringDetails>

<RR1TenantImprovements> </RR1TenantImprovements>

<RR1TenantDisrepair> </RR1TenantDisrepair>

<RR1MajorImprovement> </RR1MajorImprovement>

<ValuationCouncilTaxBand> </ValuationCouncilTaxBand>

<NoRooms> </NoRooms>

<NoBeds> </NoBeds>

<OccupationPhase> </OccupationPhase>

<RentPoints> </RentPoints>

<DiscresionaryRentPoints> </DiscresionaryRentPoints>

<MajorPhase> </MajorPhase>

<Id> </Id>

<ValuationInsurance> </ValuationInsurance>

<ValuationValue> </ValuationValue>

<PermittedNo> </PermittedNo>

<CurrentRegisteredRent> </CurrentRegisteredRent>

<TargetRent> </TargetRent>

<RentPayable> </RentPayable>

<ServicePayable> </ServicePayable>

<OtherPayable> </OtherPayable>

<SharedOwnerPercentage> </SharedOwnerPercentage>

<RR1NewRent> </RR1NewRent>

<RR1ServicesAmount> </RR1ServicesAmount>

<Valuation1999Level> </Valuation1999Level>

<ValuationVacantPosession> </ValuationVacantPosession>

<ValuationResidentialUse> </ValuationResidentialUse>

<ValuationEnergyRating> </ValuationEnergyRating>

<ValuationRateable> </ValuationRateable>

<ValuationCapital> </ValuationCapital>

<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

</PropertyRentDto>