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 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 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.

# 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. They are referred to here for completeness.

## 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

|  |  |  |  |
| --- | --- | --- | --- |
| **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

|  |  |  |  |
| --- | --- | --- | --- |
| **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

|  |  |  |  |
| --- | --- | --- | --- |
| **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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code** | **O.H.G. method** | **Civica Method Name** | **Description** | |
| **Read Services** | | | | |
| WO01 | <not requested> | GetWorksOrderDetails | | Returns works order header details |
| WO02 | <not requested> | GetWorksOrderTasks | | Returns list of tasks on works order |
| WI01 | <not requested> | GetInspectionDetails | | Returns Inspection details |
| WV01 | <not requested> | GetVisitDetails | | Returns Visit details |
| WS01 | <not requested> | GetSlotDetails | | Returns Slot details |
| **Write Services** | | | | |
| WO51 | createOrder | CollateWorksOrder | | Have system create a new rmworder record based on repair tasks on a request |
| WO52 | updateOrder | UpdateWorksOrder | |  |
| WR51 | createOrder | CreateRepairRequest | | Create a new rmreqst record |
| WR52 | createOrder | CreateRepairRequestWithTasksAndCollate | | Combines CreateRepairRequest, CreateRepairTask and CollateWorksOrder |
| WT51 | createOrder | CreateRepairTask | | Create a new rmtask record |
| WI51 | createInspection | CreateInspectionTask | |  |
| WI52 | updateInspection | UpdateInspectionTask | |  |
| WI53 |  | CollateInspectionOrder | | Have system create a new rmworder record based on inspection tasks on a request |
| WI54 | <not requested> | ProcessInspectionOutcome | |  |
| WI55 | <not requested> | CreateInspectionTaskAndProcess | |  |
| WI53 | <not requested> | DeleteInspectionTask | |  |
| WV51 | createVisit | CreateVisit | |  |
| WV52 | <not requested> | UpdateVisit | |  |
| WV53 | deleteVisit | DeleteVisit | |  |
| WV62 | updateVisitOutcome | UpdateVisitOutcome | |  |
| WS51 | createSlot | CreateSlot | |  |
| WS52 | <not requested> | UpdateSlot | |  |
| WS53 | deleteSlot | DeleteSlot | |  |

# 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>

# 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...

# UH Webservices framework

## This section is provided for info only, and summarises the webservice control and management framework that is currently being developed into the product. This framework will be used across all UH services and will be used by this project in order to provide a generic and UH standard approach. See section 2 for more info around the approach.

## Database Changes

* New table for webservice\_control
* New table for webservice\_users
* New table for webservice\_params
* New table for webservice\_audit

## Admin Console Screen

A set of new screens will be provided in UHT as a console to administer the deployed webservices. Functionality will cover:

* Maintenance of webservice controls – for enabling/disabling services; turning on/off auditing.
* Maintenance of webservice users – for adding/amending/deleting users; enabling/disabling users; linking to a UHT user account; setting/resetting user password; clearing user sessions; viewing last activity date
* Maintenance of webservice parameters – setting parameter values, such as master webservices password; session token timeout; level of authentication
* View of Audit records – with mix and match selection critieria; housekeeping options.

## Clear User Session Tokens

A routine which periodically runs which identifies any user session tokens which have been inactive for X period of time (from system parameter), and clears the session token.

***\*\*\* The above developments will be part of the standard Civica webservice framework and are detailed above for reference/information only.***

# Detailed Webservice Requirements

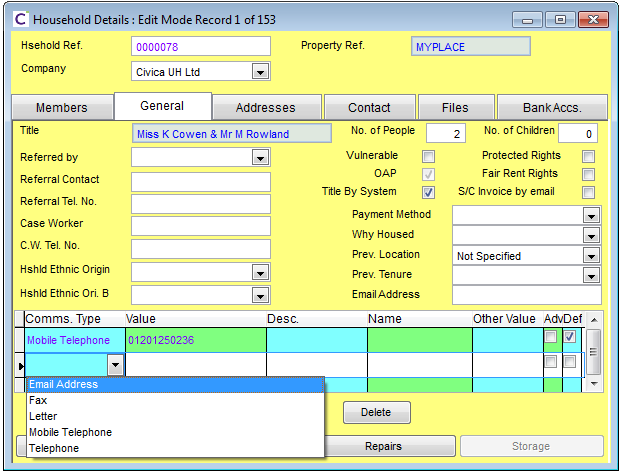
## CC51: CreateCommsEntry

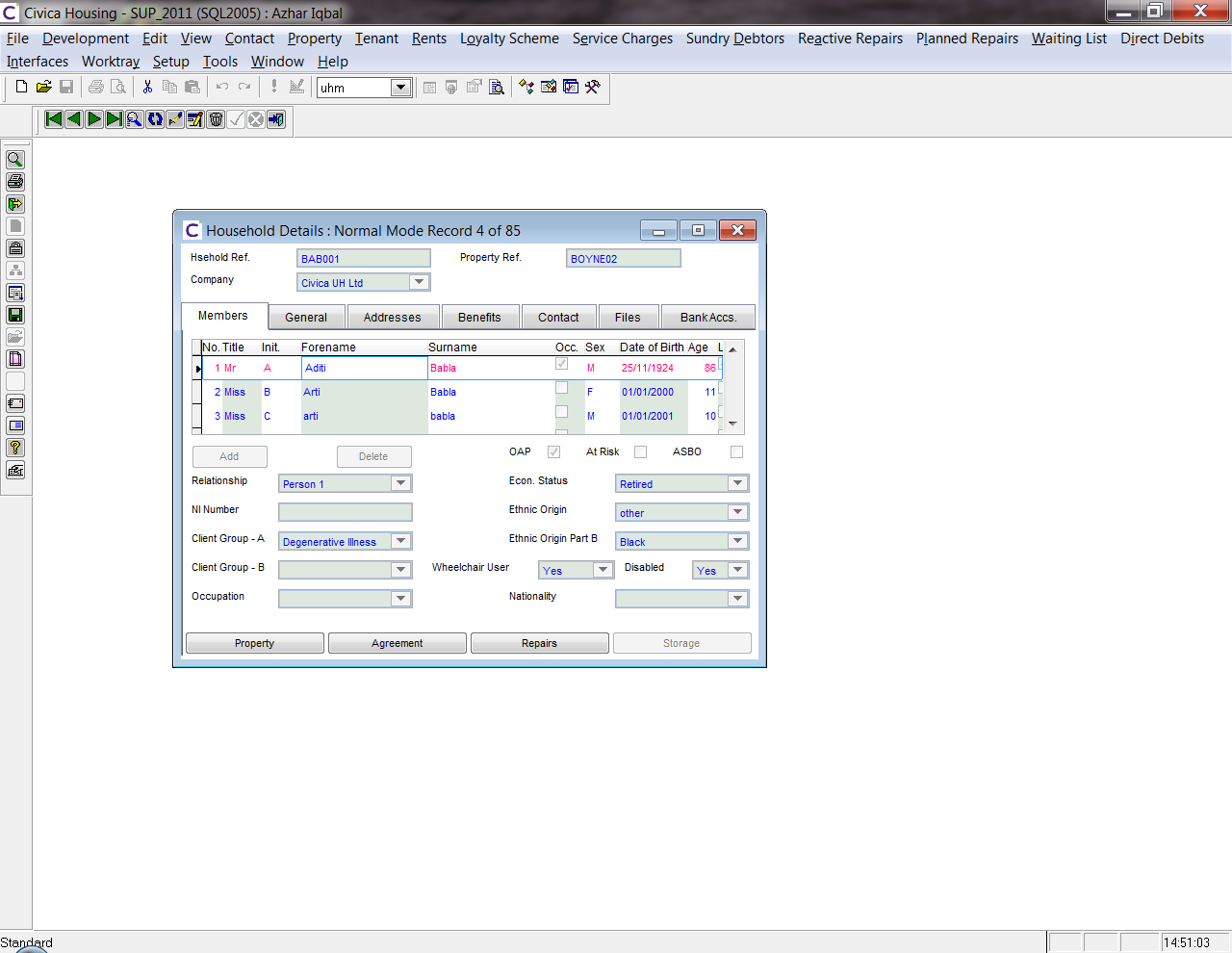
Purpose/Requirement for O.H.G:

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).

Each person within the group has a sequential reference number (1,2,3...)... It could be possible to pass in a a combined key of “Entity Key Reference + Person Sequence ID” to identify which individual a communication relates to. However, a “Name” (text) field is stored on the comms entry record, and it is proposed that the webservice includes the “Name” as an input parameter in order to identify the person. This approach replicates the functionality currently within UHT.

Note: the name field stored against the contacts is a text field, and can store any value (i.e. any name, which may or may not be one of the persons on the group members list).

A new UHT system variable will be added with a YES/NO value to determine whether the passed in Name must exist in the Person Members list for the entity (eg household).

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 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

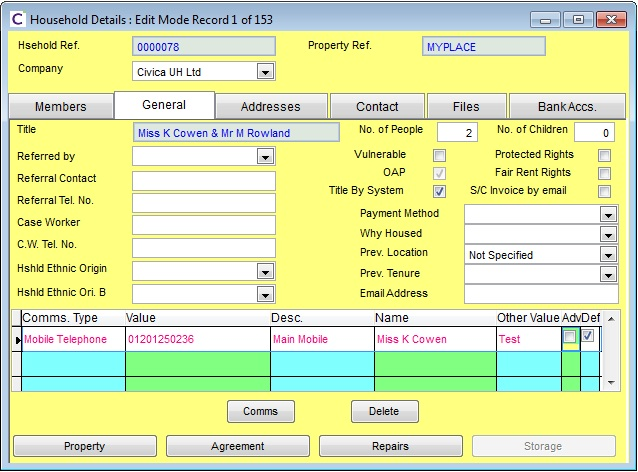
Locate the UH Comms record relating to the Comms Entry Sid, and if found delete record, else error.

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

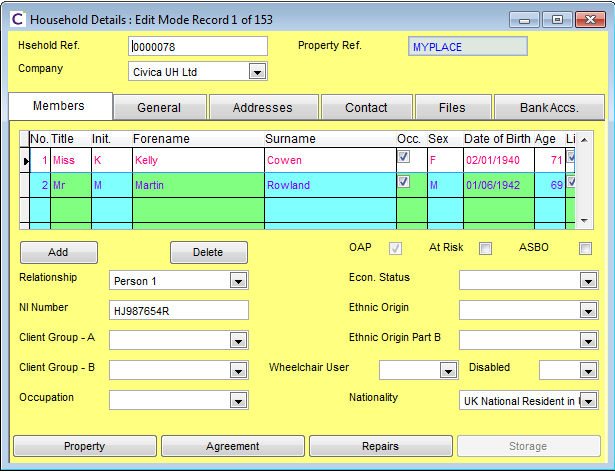
Relates to O.H.G..... also see CH62

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:

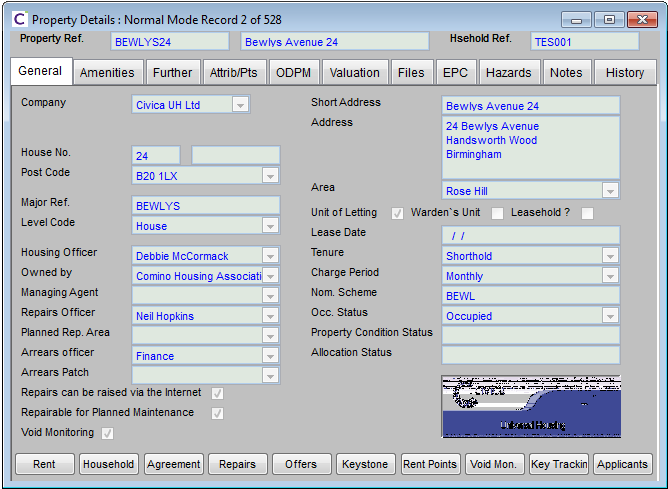
This service exists primarily to allow profiling information to be updated against a property.

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.

<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

## WO51: CollateWorksOrder

## Collates repair tasks on a request into a works order as per the existing UHT business logic

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
  + String: RepairRequestReference
* Response
  + Reference or the works order created
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Repair tasks on the request specified that have not been collated into an order yet will be collated following the existing UHT business logic

Errors

TBA

## WO52: UpdateWorksOrder

This service is to facilitate the update of Works Order records, following the existing UHT business logic

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: WorksOrderDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Validation will be performed to ensure read only fields are not altered. WorksOrderDto will have a key of wo\_ref \ WorksOrderReference. The matching record in rmworder will be updated as per the existing UHT business logic. If there is no such record an error will be returned.

Errors

TBA

## WR51: CreateRepairRequest

This service is to facilitate the creation of repair request records following the existing UHT business logic

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: RepairRequestDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Validation will be performed to ensure fields not available to the user in UHT are not set on the DTO. A new repair request record will be created as per the existing UHT business logic.

Errors

TBA

## WT51: CreateRepairTask

This service is to facilitate the creation of new a repair task

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
  + String: RequestReference
  + Data Object: RepairTaskDto
    - (see Appendix A)
* Response
  + Sreinf: TaskReference
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

A new rmtask record will be created based on the DTO and linked to the specified repair request record. The generated task\_ref of the new record will be returned.

Errors

TBA

## WR52: CreateRepairRequestWithTasksAndCollate

This service is to facilitate the creation of Works Order records, according to the requirements of O.H.G.  
  
The specification below needs further analysis and review and is therefore subject to change.

Background

In UHT, repair requests are raised which creates an rmreqst record on the “Diagnose Repair Form” (Reactive repairs menu). The user will select the property the repair is to be carried out at and add tasks (rmtask records) that require to be completed. Upon the print order button being selected, the system will “Collate” these tasks into a “Works Order” (rmworder record). Depending on system settings all the tasks may be collated onto a single order or they may be split onto multiple orders. There may be a “Minimum order” value configured which represents the minimum value that should be on an order. If the value of the tasks doesn’t add up to that amount a “dummy” task is added to the request \ order with a value to make the order up to the minimum allowable.

Webservice Requirements / Assumptions

* 1:1 relationship between request and works order (i.e. a request only ever has one works order).
* The priority for each task on the works order is the same.
* Unless specified on the tasks passed in to the webservice, the trade should be derived from the SOR
* Unless specified by the data passed in to the webservice, the due date should be calculated from the priority.

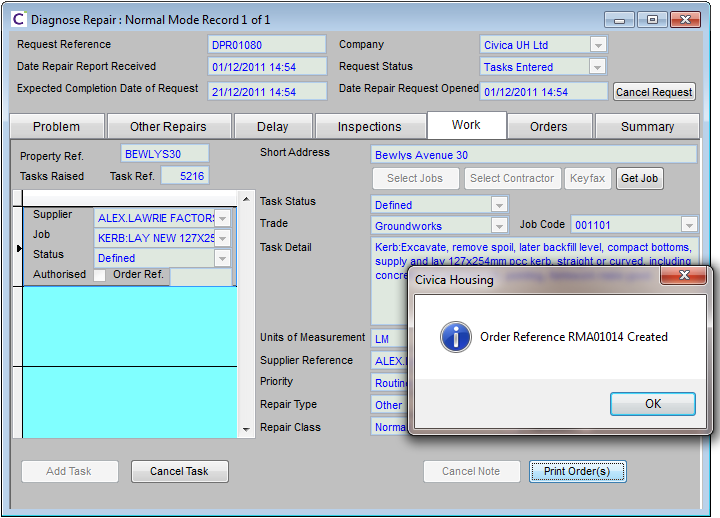
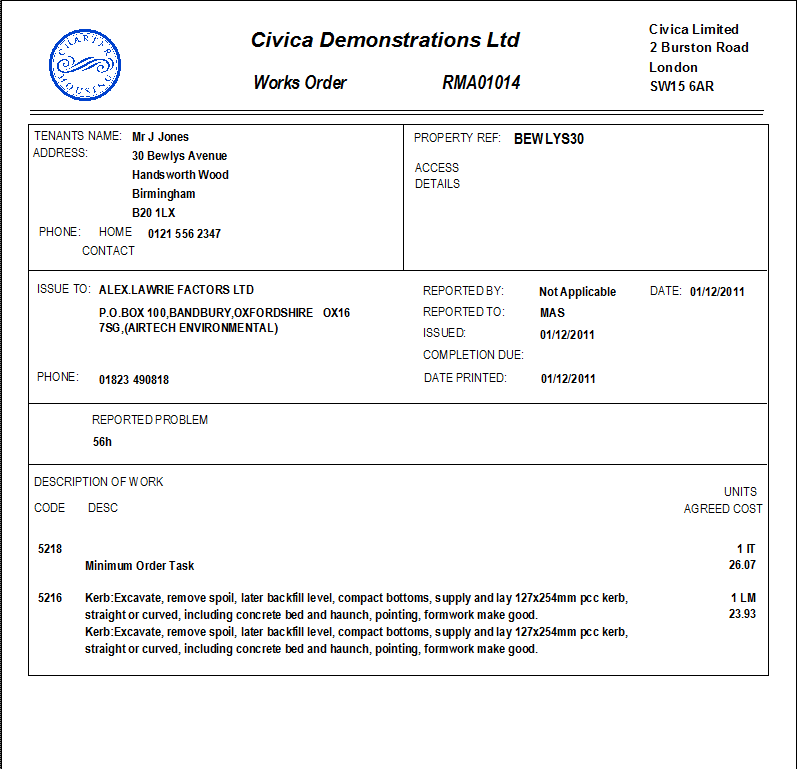
O.H.G. have requested:  
  
*“It should be possible to adjust the cost of the order independent of the tasks (the cost of which we do not require setting) using the total\_cost. This should be achieved by the use of MINVAL and REDUCE tasks. These tasks should be invisible, however, in that they are never returned as part of the collection of tasks and attempts to update them using the sequence number will fail.*

*The MINVAL / REDUCE task should have a sequence number of 0, -1 or a very large number to prevent them colliding with inputted sequence numbers.”*

Civica:

* We need to clarify the concept of the “REDUCE” task, which the UHT development team are not familiar with. We presume this is proposed new functionality where a dummy task will be added with a negative value to artificially alter the cost at the order level and satisfy the business rule of an orders cost being equal to that of the sum of the tasks ???
* Currently we are not aware of negative value tasks being used in UHT and whether this is permissible with the current business logic within UHT.

Action is required to clarify the above points, and validate the proposed solution for feasibility.

Separate methods will be created to replicate the UHT functionality of creating a repair request, adding a repair task to a repair request and collating the tasks on a request into an order. A method will be created for OHG that combines these three processes so that they can be achieved in a single web service call that takes a NewRepairDto as a parameter. This is in turn made up of a RepairRequestDto and a collection of RepairTaskDto  
  
  
  
  


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: NewRepairDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)
  + Wo\_ref (works order reference)

Processing

Creation of repair request record in UHT table: rmreqst along with associated task records in table: rmtask. Creation of works order record on table: rmworder.

Errors

TBA

## WI51: CreateInspectionTask

This service is to facilitate the creation of a new inspection task

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: InspectionTaskDto
    - (see Appendix A)
* Response
  + String: Task Reference
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Similar to WT51 but an inspection task will be created rather than a regular repair task. The normal UHT business logic will be respected.

Errors

TBA

## WI52: UpdateInspectionTask

## This service is to facilitate the update of an inspection task

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: InspectionTaskDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

The key of the InspectionTaskDto will be task\_ref \ TaskReference. This rmtask record will be updated with the changes specified in the DTO. Validation will be performed to ensure read-only fields do not get updated. The existing UHT business logic will be followed. If there is no such record in rmtask a SOAP fault will be returned.

Errors

TBA

## WI53: CollateInspectionOrder

## This service is to facilitate the collation of Inspections tasks on the specified repair request into an inspection order

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
  + String: RequestReference
* Response
  + String: WorksOrderReference
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Inspection tasks on the specified repair request that have yet to be collated onto an order will be collated as they would if the “Print Inspection(s)” button were clicked on the Inspections tab of the Diagnode repair form.

Errors

TBA

## WI54: ProcessInspectionOutcome

## This service is to facilitate the completion of an inspection order

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: InspectionOutcomeDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

The key of the InspectionOutcomeDto will be wo\_ref \ WorksOrderReference. The specified works order will be completed with the specified inspection outcome details. This will mirror the UHT functionality involked by the “Confirm Action” from the “Inspection Outcome” form.

Errors

TBA

## WI55: CreateInspectionCollateAndProcess

This service is to facilitate the creation of Works Inspection records, according to the requirements of O.H.G.  
  
The specification below needs further analysis and review and is therefore subject to change.

Background

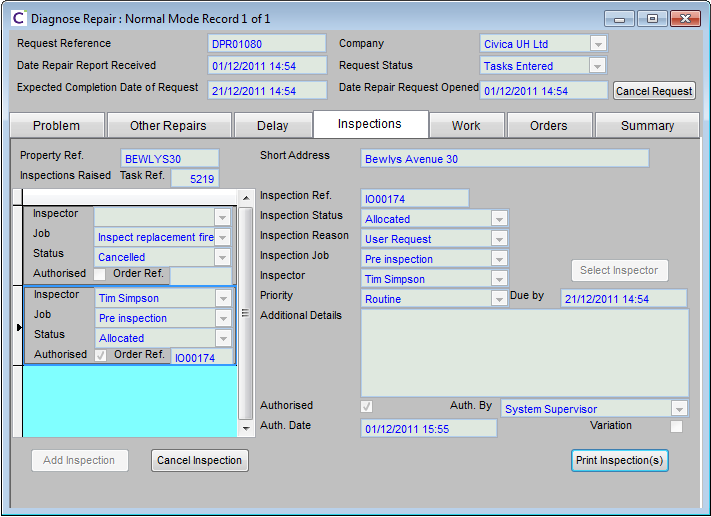
In UHT: Pre, Post and WIP Inspections can be raised. Inspections consist of inspection tasks and inspection orders just like normal repairs and are stored against the same tables – rmtask \ rmworder. In UHT the tasks are added first and when the print inspections button is pressed they get collated into an inspection order by the system.

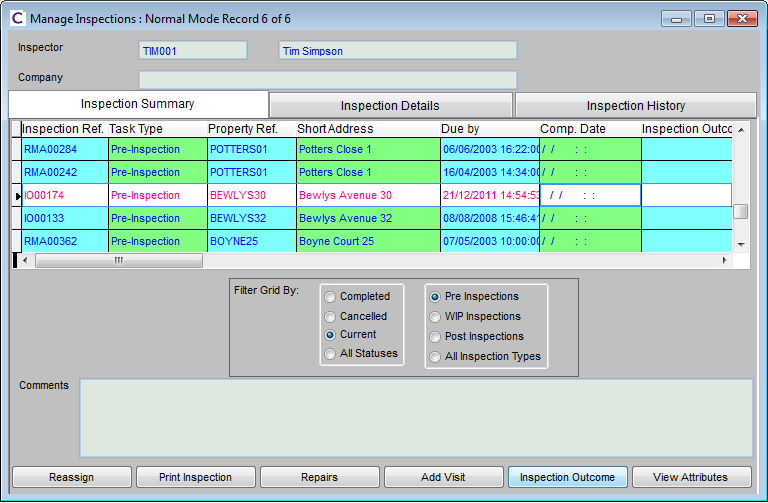
The separate functions will be written for the 3 processes of adding an inspection task, collating the order and completing the inspection which mirror the UHT functionality. This service will call the 3 fuctions one after another so the client can achieve all steps in a single service call.

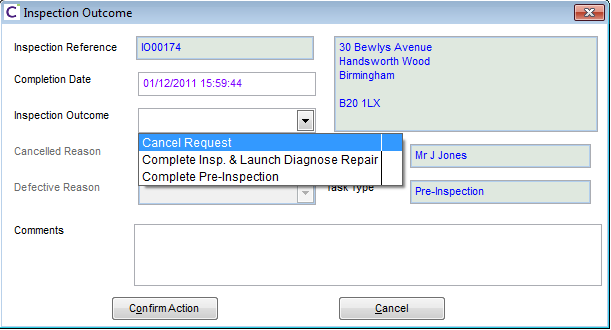
O.H.G. have requested that the “Inspection narrative” and “post Inspection result” free text boxes in the workflow processes would be useful additions if they write into UHT.

Webservice Requirements / Assumptions

* The Inspection order should be associated with a repair request, from which the priority should be derived. (Note: the OHG spec infers that the Inspection should be linked to a works order – in UHT they are linked to their parent repair request).







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: RepairTaskDto, InspectionOutcomeDto(see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)
  + Inspection\_Wo\_ref (inspection order reference)

Processing

Creation of inspection record in UHT table: rmworder along with associated task records in table: rmtask.

Errors

TBA

## WI52: UpdateInspectionTask

This service is to facilitate the update of Works Inspection records, and inspection outcomes according to the requirements of O.H.G.  
  
The specification below needs further analysis and review and is therefore subject to change.

Note: There can be logic of the back of an inspection completion. E.g. if the inspector decided no work needs to be done - the request may be cancelled.

Webservice Requirements / Assumptions

* See WI51

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: InspectionTaskDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Update of inspection task record in UHT table: rmworder along with associated task records in table: rmtask.

Errors

TBA

## WT51: UpdateInspectionOrder

## This service is to facilitate the updating of an inspection order

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
  + String: RequestReference
  + Data Object: InspectionOrderDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

The key of the InspectionOrderDto will be wo\_ref \ WorksOrderReference. The related record in rmworder will be updated as per the existing business logic in UHT. If there are any changes to read-only fields a SOAP fault will be returned. If there is no related record in rmworder again a SOAP fault will be returned.

Errors

TBA

## WI53: DeleteInspectionTask

This webservice is to facilitate the deletion of Inspection task records.

<not required for O.H.G.>

## WV51: CreateVisit

This service is to facilitate the creation of Visit records, according to the requirements of O.H.G.  
The specification below needs further analysis and review and is therefore subject to change. The OHG spec suggests a link between visits and diary maintenance. Currently there is no such link in UHT. Need to confirm with OHG exactly how they want this to work.



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: WorksVisitDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)
  + Visit\_Sid (unique reference)

Processing

TBA

Errors

TBA

## WV52: UpdateVisit

This webservice is to facilitate the update of Visit records.

<not required for O.H.G.>

## WV53: DeleteVisit

This service is to facilitate the removal of a vist, according to the requirements of O.H.G.  
  
The specification below needs further analysis and review and is therefore subject to change.

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
  + Visit\_Sid
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

This service should clear (zero-ise) all values pertaining to the visit, rather than deleting the slot, thereby freeing it to be used again.

Errors

TBA

## WV62: UpdateVisitOutcome

This service is to facilitate the update of a visit outcome, according to the requirements of O.H.G.  
  
The specification below needs further analysis and review and is therefore subject to change.

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: VisitOutcomeDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Update the visit outcome and associated comments.

Errors

TBA

## WS51: CreateSlot

This service is to facilitate the creation of slots into the diary, according to the requirements of O.H.G.  
  
The specification below needs further analysis and review and is therefore subject to change.

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: SlotsDto
    - (see Appendix A)
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

Creation of a new slot.

Errors

TBA

## WS52: UpdateSlot

This webservice is to facilitate the update of Slot records.

<not required for O.H.G.>

## WS53: DeleteSlot

This service is to facilitate the deletion of a slot from the diary, according to the requirements of O.H.G.  
  
The specification below needs further analysis and review and is therefore subject to change.

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
  + Visit\_Sid
* Response
  + SOAP response for success/failure (SOAP-ENV fault tag)

Processing

If the slot is booked, then it cannot be deleted and the service should fail with an error code.

Otherwise, the record should be deleted.

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. but some are summarised below. The list will be extended during future design.

* 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>

## NewRepairDto

\*\* to be confirmed

<NewRepairDto>

<RepairRequestDto>

<See 9.8>

<RepairRequestDto>

*collection of tasks:*

*<Tasks>*

*<RepairTaskDto>*

<see 9.7>

<RepairTaskDto>

</Tasks>

<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

</NewRepairDto>

## RepairTaskDto

At a minimum will represent the task related fields available to the user on the Diagnose Repair form.

\*\* to be confirmed

<RepairTaskDto>

<task\_ref> </task\_ref>

<sup\_ref> </sup\_ref>

<job\_code> </job\_code>

<trade> </trade>

<task\_text> </task\_text>

<unit\_narr> </unit\_narr>

<priorirty> </priority>

<rep\_type> </rep\_type>

<rep\_class> </rep\_class>

<est\_units> </est\_units>

<est\_code> </est\_cost>

<date\_due> </date\_due>

<rq\_location> </rq\_location>

<variation> </variation>

<authorised> </authorised>

<wo\_ref> </wo\_ref>

<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

</RepairTaskDto>

## RepairRequestDto

At a minimum will represent the fields available to the user on the Diagnose Repair form.

<RepairRequestDto>

*details TBA ... to include*

<rq\_ref> </rq\_ref>

<prop\_ref> </prop\_ref>

<rq\_status></rq\_status>

<rq\_phone><rq\_phone>

<rq\_problem><rq\_problem>

<rq\_priority></rq\_priority>

<UserFields>

<KeyValueOfStringString>

<Key>u\_myuserfield</Key>

<Value> ValueToSet</Value>

</KeyValueOfStringString>

</UserFields>

<RepairRequestDto>

## InspectionOutcomeDto

At a minimum will represent the fields available to the user on the Inspection Outcome form.

<RepairRequestDto>

*details TBA ... to include*

<CompletionDate></Completion\_date>

<InspectionOutcome></InspectionOutcome>

<Comments></Comments>

<RepairRequestDto>

## WorksVisitDto

<WorksVisitDto>

*details TBA ... to include*

<wo\_ref>

<hadiary\_sid>

<visit\_prop\_appointment>

<visit\_slot\_type>

<visit\_outcome>

<visit\_comment>

</WorksVisitDto>

## VisitOutcomeDto

<VisitOutcomeDto>

*details TBA ... to include*

<visit\_sid>

<visit\_outcome>

<visit\_comment>

</VisitOutcomeDto>

## SlotsDto

<SlotsDto>

*details TBA ... to include*

<hadiary\_sid> <hadiary\_sid>

<visit\_prop\_appointment> <visit\_prop\_appointment>

<visit\_slot\_type> <visit\_slot\_type>

<SlotsDto>