



Let's go further



EG MainManager FM

Technical description

Revision: 1.5

Date: 13.07.2022

Contents

1.	Introduction.....	4
1.1.	System purpose	4
1.2.	High level system description.....	4
2.	System architecture.....	4
3.	System description	6
3.1.	Development platform	6
3.2.	Security.....	7
3.2.1.	Authentication	7
3.2.2.	System access	7
3.2.3.	Data access	7
3.3.	Database.....	7
3.3.1.	Database per customer.....	7
3.3.2.	Azure.....	7
3.3.3.	Other.....	7
3.4.	File storage	8
3.4.1.	Azure.....	8
3.4.2.	Other.....	8
3.5.	Logging	8
3.6.	Integration options.....	9
3.6.1.	Message based integration framework	10
3.6.2.	Action queue based integration.....	11
3.7.	Standard integration	12
3.7.1.	Authentication integrations.....	12
3.7.2.	Official Asset-Data integration	12
3.8.	Data import/migration options	13
3.9.	Scalability (customer hosted systems)	13
3.10.	Service/component (when not hosted in Azure)	13
3.10.1.	Web applications	13
3.10.2.	Windows Services	13
3.11.	Functional dependencies.....	14
3.12.	Network.....	15
3.13.	General items	15
4.	Further information.....	15
5.	Appendix	17
5.1.	Web service API.....	17
5.1.1.	REST API	17
5.1.2.	Asset information SOAP API.....	19

5.1.3.	Main processes SOAP API	20
5.2.	Asset data integrations.....	21
5.2.1.	BBR in Denmark	21
5.2.2.	Matrikkel in Norway	21
5.2.3.	Fasteignamat in Iceland	21
5.3.	Migration methods for Assets.....	21
5.3.1.	Import Structure\Sites (Ground).....	22
5.3.2.	Import Structure\Region.....	22
5.3.3.	Import Structure\Real Property.....	22
5.3.4.	Import Structure\Buildings	22
5.3.5.	Import Structure\Building Space.....	23
5.4.	Migration methods for Organizations and Human resources.....	23
5.4.1.	Import Organizations\Company.....	23
5.4.2.	Import Organizations\Employees.....	24
5.5.	Import methods for Plans	24
5.6.	Migration and integration methods for Finance management.....	24
5.6.1.	Import Accounts\Accounts	24
5.6.2.	Export Accounts\Accounts.....	28
5.6.3.	Import Finance\Transactions	29
5.6.4.	Import Contracts\Lease Contracts	31
5.7.	Integration methods for Operation management	32
5.8.	Integration methods for Service management	32
5.9.	Integration methods for Administration	33
5.10.	Integration API - Uncategorized	33
5.11.	Integration with GIS systems.....	41

1. Introduction

1.1. System purpose

MainManager is a comprehensive Facility Management system that will simplify and coordinate management of the organizations facilities.

1.2. High level system description

Figure 1 shows the high level description of a standalone version of the system.

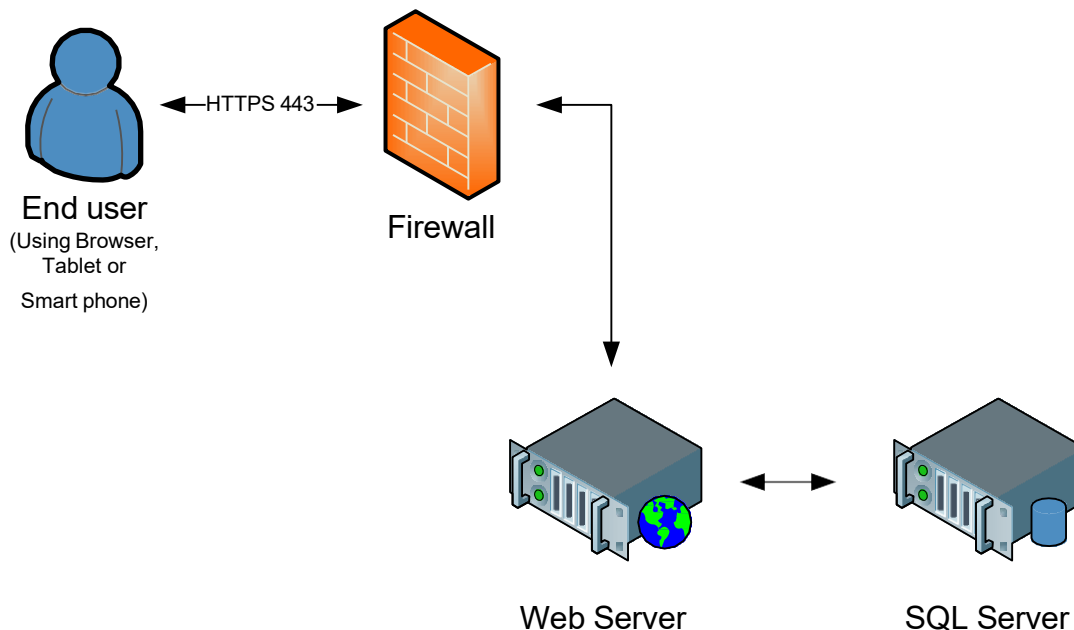


Figure 1. High level system.

2. System architecture

The system is based on ASP.NET infrastructure and is a web application with an exposed REST API for integrations. The following figure shows an overview of the system architecture and key components.

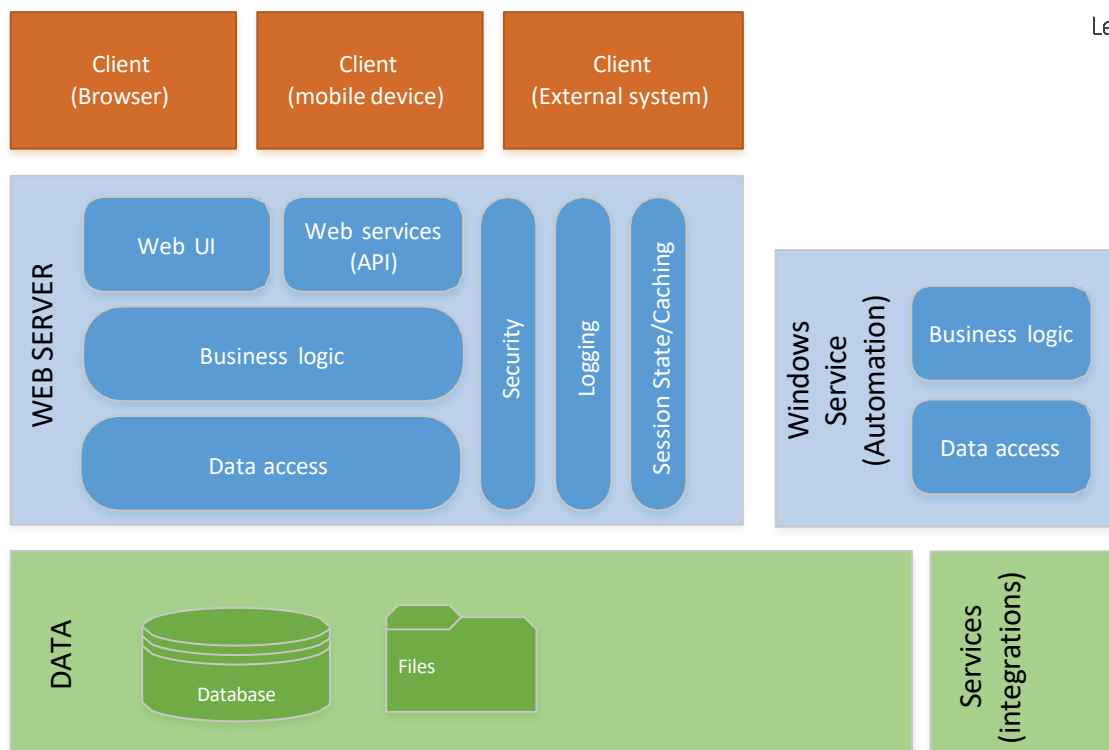


Figure 2. System architecture.



Figure 3. Logical components of the web solution.

3. System description

3.1. Development platform

MainManager is a Web Application, accompanied by Windows Service for automation, built using Microsoft .NET framework 4.7 and runs on Microsoft Windows 2012/2019 servers. The software is developed using Microsoft Visual Studio 2019 development environment and the development is managed using SCRUM methodology.

The MainManager App is developed using Phone gap, a mobile development environment for hybrid app development. Current tested mobile platforms are Android and iOS (Apple devices).

3.2. Security

The system supports SSL/TLS (HTTPS) configuration. When the system is published to the internet, it is recommended that the customer installs such a certificate on the respective web server(s) and configure the system to require SSL/TLS.

3.2.1. Authentication

MainManager supports various authentication scenarios:

- Form authentication
- Microsoft Active Directory integration
 - AD – login only
 - AD – Single-sign-on only
 - AD – Single-sign-on and external login
- SAML 2 - Single-sign-on
- AD single sign on for hosted customers
- Standard login with Azure AD single-sign-on authentication
- Standard login with Google single-sign-on authentication

To ease the user management, the use of integrated user management and authentication is recommended. The system provides API for this purpose and supports synchronising with Active Directory or Azure AD.

See MainManager Security Guide for more information.

3.2.2. System access

System access is controlled by the user role. The user role specifies which system entities the respective user can access, and the access level (read/change/create/delete).

3.2.3. Data access

Data access is determined by assigning the respective employee, or parent organisation, access to respective data entities, typically site groups.

3.3. Database

3.3.1. Database per customer

Each customer system has its own database. This ensures high data security.

3.3.2. Azure

The Azure SQL Database is used.

3.3.3. Other

The system uses Microsoft SQL Server as a data repository. Versions 2016 and newer are supported. If the customer is hosting the system on-prem the system can use existing shared SQL Server if that is the customer preference, but for larger systems a dedicated server is recommended.

3.4. File storage

3.4.1. Azure

Azure Blobs are used for file storage. Each customer has its own blob container and can only read/write to that container. This ensures high file security.

3.4.2. Other

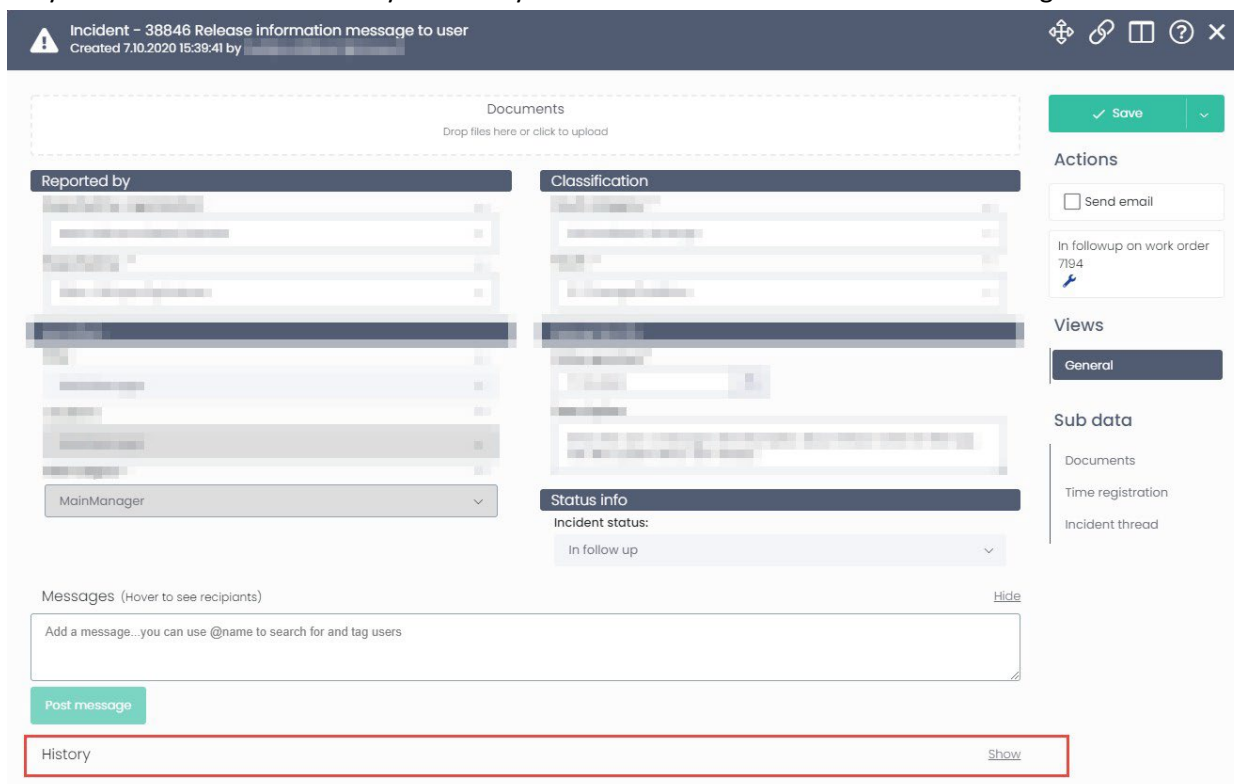
Files are stored on a file folder/share. Each system has its own folder and is only able to read/write to that folder. This ensures high file security.

3.5. Logging

The system logs the following user activity within the system:

- Logins
- Record interaction, create, view, edit, delete
 - Who did what and when
- Actions runs

The log is stored in the system database and users can see the access and change history of the records they have access to within the system. Only administrators have access to the entire log.



Incident - 38846 Release information message to user
Created 7.10.2020 15:39:41 by [redacted]

Documents
Drop files here or click to upload

Reported by

Classification

Save

Actions

☐ Send email

In followup on work order 7194

Views

General

Sub data

Documents

Time registration

Incident thread

Messages (Hover to see recipients)

Add a message... you can use @name to search for and tag users

Post message

History

Figure 4. Access to the log for a record.

History		Hide
[38846]	Viewed	
8.10.2020 15:41:50		
Created		
The following table shows the history of the record. The table contains the date and time when the record was created, the user who created the record, and the user who viewed the record.		
7.10.2020 15:39:42		

Figure 5. History log for the record.

3.6. Integration options

MainManager is designed to integrate to multiple systems. To ensure single master catalogue for each type of record, integration can be used to enable automatic and consistent information flow between systems. In the spirit of Facility Management, MainManager is designed to be a master catalogue for property information and some related information, and integration to other systems ensure effective data exchange and ensures data integrity within the respective organisation. As an example of such integration is the following figure.

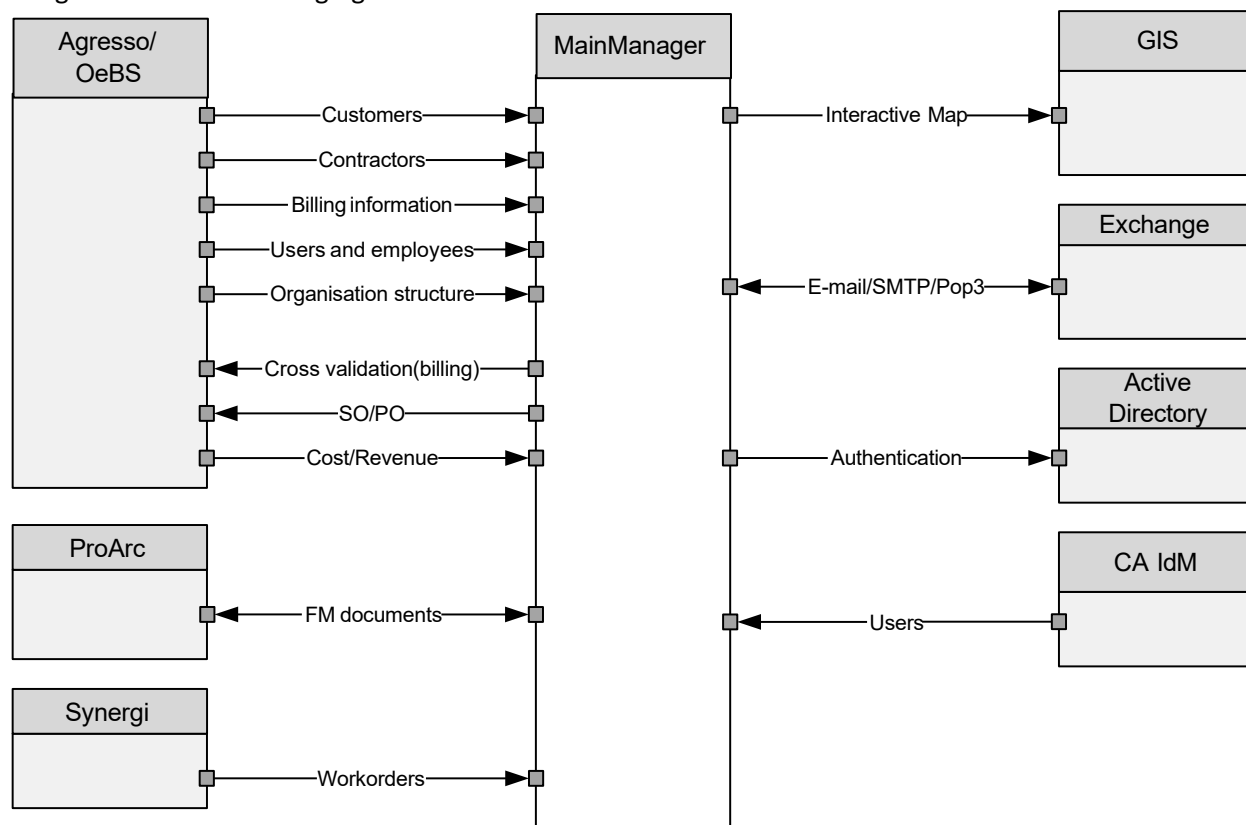


Figure 6. An example of system integration.

MainManager provides SOAP and REST Web service API that can be used for integration by other systems. The system is developed using open standards for data storing and is open for customized integration to systems supporting open standards or proprietary integration when .Net integration library is provided. Following integration techniques are now used in the system:

- Synchronous SOAP web service
- Synchronous REST/Json web services (preferred)
- Asynchronous file import from FTP/SFTP
- Synchronous SQL/ODBC direct database integration (TCP)
- Asynchronous message based Oracle Message Queue (using Weblogic JMS Queue (XML)) with configurable triggering fields
- Action queue for publication of object changes with configurable triggering fields
- Asynchronous email, Pop3/SMTP
- Synchronous LDAP authentication
- Synchronous SAML2 authentication

Integrations are typically performed using actions (functions). These actions can be run manually or, if appropriate can be configured to be run periodically according to configured schedule. Importing is implemented using an import/update methodology, i.e. new items are created but items that already exist are verified and updated if needed. Typically it can be configured whether import actions only import new items or both import new items and updates existing items.

Import report is generated each time an action is run. When run manually, the results and import report is the result window. Typically, integration actions can be configured to send integration report to configured email addresses. See section 5.1 for further information.

3.6.1. Message based integration framework

The system has a framework to support message based integration (message queues). Following figure shows the incoming queue processing.

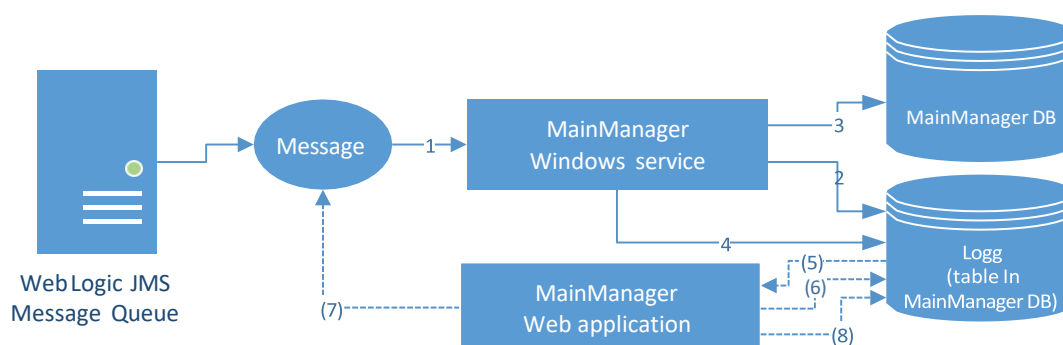


Figure 7. Incoming messages.

1. Queue service listens (polls) and retrieves messages from the queue.
2. Waiting messages stored in log table.

3. The service processes the unhandled messages. If message timestamp is older than datechanged timestamp on the destination record, the message is not used and it is marked as deleted. Otherwise the record is updated in MainManager.
4. The service marks the message as processed, error or deleted. See error handling for more information.
5. Messages market as failed can be pushed to the incoming queue for reprocessing. This is a manual action. The message can also be processed directly from the queue.
6. Message is logged.
7. Message is pushed to queue.
8. Message is marked as processed or on error. The action shows errors in result window.

Following figure shows the event based outgoing queue.

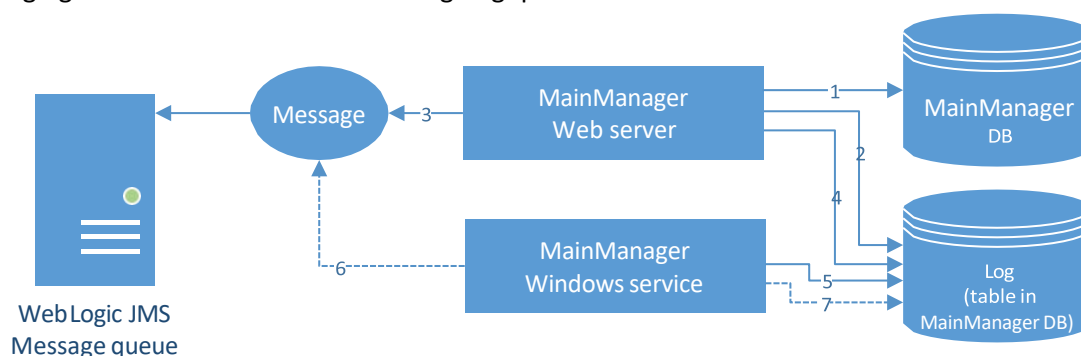


Figure 8. Outgoing queue, event driven.

1. Change that triggers a message.
2. Message created and logged
3. Message pushed to queue
4. Message market as handled (message pushed to queue without error) or error (could not be pushed to queue successfully).
5. The service checks for messages market on error in the log.
6. If there are messages market on error the service will try to push them to queue.
7. If the messages can be successfully pushed to queue, then they are market as processed in log.

The outgoing messages can also be processed periodically instead of event based.

3.6.2. Action queue based integration

Part of the above message based framework is also used as fail safe, configurable integration. Instead of an XML message, an action is triggered to run when one of the configured fields changes. This action is configurable and is typically a web hook implementation, i.e. a call to a web service in a central integration platform that is registered as a subscriber to add/change/delete events in the system. If, for some reason, the subscriber call fails, the automation will retry the sending for specified amount of time. The following figure shows a sequence diagram of an asset publication integration using this framework.

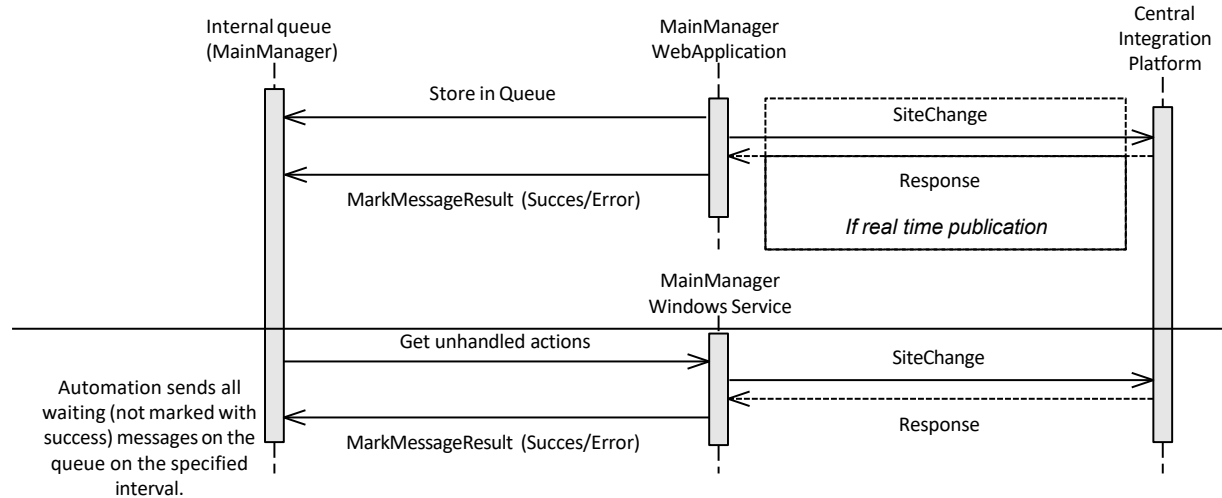


Figure 9. Example of queue based integration using internal message/action queue.

When this is used the central integration platform provides a web service to receive the required entities and fields which MainManager calls using a customized action.

As detailed in Appendix 5.1, MainManager provides subscription API for add/change/delete events for asset information (sites, buildings, building floors and spaces) using this infrastructure.

3.7. Standard integration

MainManager supports some useful integrations out of the box. Here are some of the most used integrations.

3.7.1. Authentication integrations

MainManager supports several authentication integrations to enable customers to use single user management system. Please see MainManager - System Security Guide.pdf for further information.

3.7.2. Official Asset-Data integration

It is useful to have assets registered according to official records. Therefore MainManager has standard integration to some data sources that provide this information.

- BBR in Denmark
- Matrikkel in Norway
- Fasteignamat in Iceland

Following sections describe these integration options.

BBR in Denmark

Matrikkel in Norway

Fasteignamat in Iceland

3.8. Data import/migration options

The system has variety of migration and import options. These are typically used for data migration in new implementations and also periodically for e.g. yearly budget. Custom migration method can be added if no existing method is suitable.

3.9. Scalability (customer hosted systems)

The system can be scaled to support thousands of simultaneous users. Following figure shows an example of a configuration that supports up to 2000 simultaneous users.

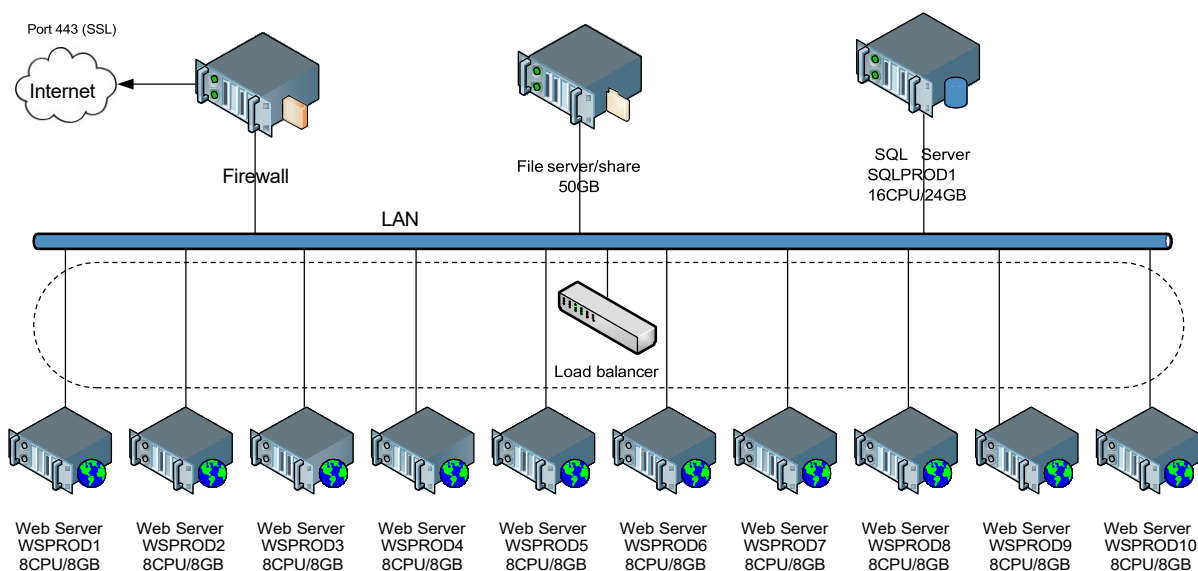


Figure 10. Recommended server configuration for up to 2000 simultaneous users.

See MainManager - Server Resource Selection Guide for more detailed techniques.

3.10. Service/component (when not hosted in Azure)

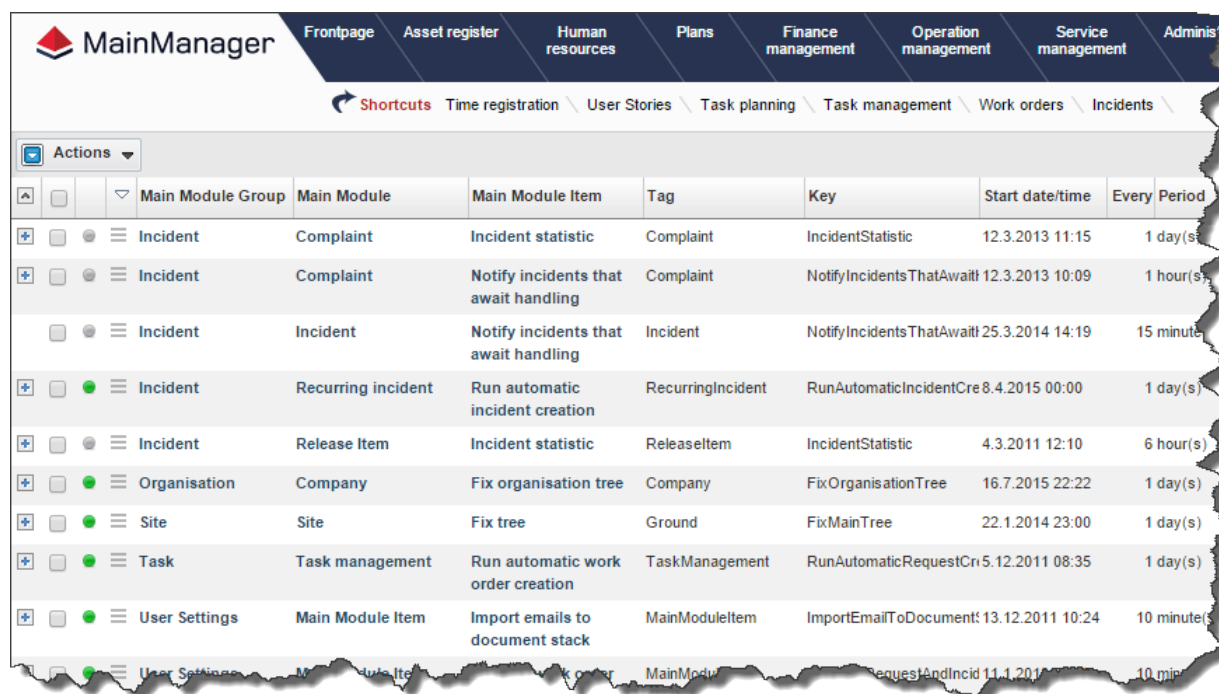
3.10.1. Web applications

MainManager web application is installed on one or more web servers.

3.10.2. Windows Services

MainManager windows service is installed on one server.

The service runs all configured automatic actions according to the configured schedule. If the respective action fails to run successfully, email is sent to the email(s) configured in the automatic actions global settings. The automatic actions can be configured and monitored in the Administration window.



Main Module Group	Main Module	Main Module Item	Tag	Key	Start date/time	Every	Period
Incident	Complaint	Incident statistic	Complaint	IncidentStatistic	12.3.2013 11:15	1 day(s)	
Incident	Complaint	Notify incidents that await handling	Complaint	NotifyIncidentsThatAwait	12.3.2013 10:09	1 hour(s)	
Incident	Incident	Notify incidents that await handling	Incident	NotifyIncidentsThatAwait	25.3.2014 14:19	15 minute(s)	
Incident	Recurring incident	Run automatic incident creation	RecurringIncident	RunAutomaticIncidentCre	8.4.2015 00:00	1 day(s)	
Incident	Release Item	Incident statistic	ReleaseItem	IncidentStatistic	4.3.2011 12:10	6 hour(s)	
Organisation	Company	Fix organisation tree	Company	FixOrganisationTree	16.7.2015 22:22	1 day(s)	
Site	Site	Fix tree	Ground	FixMainTree	22.1.2014 23:00	1 day(s)	
Task	Task management	Run automatic work order creation	TaskManagement	RunAutomaticRequestCr	5.12.2011 08:35	1 day(s)	
User Settings	Main Module Item	Import emails to document stack	MainModuleItem	ImportEmailToDocument	13.12.2011 10:24	10 minute(s)	
User Settings	Main Module Item	Import emails to document stack	MainModuleItem	ImportEmailToDocument	11.1.2011 10:24	10 minute(s)	

Figure 11. The automatic action window.

Integration actions can run successfully even though errors occur. Such services can typically be configured to send result emails containing report including respective errors if they occur.

3.11. Functional dependencies

The system can depend on other systems and services. Following table shows possible general functional dependencies.

Function	Purpose	Comment
SMTP	Email sending	
POP3	Mailbox reading	
Active Directory Sign on	To enable single sign on and sign on by using AD user/pass.	This feature performs lookup in active directory on each login. The user name in MainManager must match the AD username. See Configuration Manual for detailed description of different AD configurations.
SAML2	Authentication and single sign on	This provides single-sign-on and single-log-out service.
BIM	BIM usage within MainManager	Only if the system is configured for BIM usage
Service and documentation	Translation updates and user role management. On line documentation.	
SMS(mobile text message)	SMS (mobile text message) sending	

3.12. Network

The system needs to have network access to other systems and services. Following table shows possible general functional dependencies.

Service/Component	Port#	Source	Destination	Comment
SQL - [Env]	TCP 1433	[SRVR]	[SQL server]	
Active Directory	TCP 389 or TCP 636 (SSL)	All servers	web [LDAP server] or [LDAP server](SSL)	LDAP - Lightweight Directory Access Protocol LDAPS - Lightweight Directory Access Protocol over TLS/SSL
Email (outgoing)	TCP 25	All servers	web [SMTP server]	SMTP for e-mail integration
Email (incoming)	TCP 110	N/A	N/A	Each server can read from arbitrary number of mailboxes.
Proxy server	TCP ?	All servers	web [Proxy server]	If the customer uses proxy servers for external website access, this must be set in Web.config files on all MainManager web servers.
BIM	TCP 443	All servers	web https://api.bimsync.com/	If the system is configured for BIM this path must be open
Documentation			documentation.mainmanager.com	
Service			service.mainmanager.is	Provides user role library and translations.
App	TCP 80	Mobile devices	[web]/services/appprocesses.sasmx [web]/services/appprocesses.sasmx/SaveImage	In case of firewall openings, these paths have to be opened for apps to function

3.13. General items

Item	Information	Comment
Log files	C:\Log	Common log file folder for web and windows service (automatic services)
Monitoring	Customer monitoring system	Used for monitoring of MainManager web application, database and services.

4. Further information

If further information is needed please contact your Main Manager contact, send an inquiry to support@mainmanager.com or call us at +354 412 8600.



Let's go further

Any suggestions for improving this manual are appreciated, please send to support@mainmanager.com.

5. Appendix

5.1. Web service API

5.1.1. REST API

From version 11.7 new API is developed as REST web services. The Swagger documentation framework is used for ease of documentation and tests. Here is the link to the newest API documentation For MainManager FM:

<https://servicemmfmm.mainmanager.is/swagger>

Following image shows an overview of the existing API at the time of writing.

AccountingV1			Show/Hide	List Operations	Expand Operations
GET	/api/v1/accounts	The operation gets multiple accounts from the MainManager API.			
POST	/api/v1/accounts	The operation creates accounts with the MainManager API.			
PUT	/api/v1/accounts	The operation creates or update account with the MainManager API.			
GET	/api/v1/accountslookup				
GET	/api/v1/financetransactions	The operation gets multiple finance transactions from the MainManager API.			
POST	/api/v1/financetransactions	The operation creates finance transaction with the MainManager API.			
PUT	/api/v1/financetransactions	The operation creates or update finance transaction with the MainManager API.			
GET	/api/v1/workorders	The operation gets multiple workorders from the MainManager API.			
POST	/api/v1/workorders	The operation creates work order with the MainManager API.			
PUT	/api/v1/workorders	The operation creates or update work order with the MainManager API.			
GET	/api/v1/financedatasets	The operation gets multiple finance dataset from the MainManager API.			
POST	/api/v1/financedatasets	The operation creates accounts with the MainManager API.			
PUT	/api/v1/financedatasets	The operation creates or update financedataset with the MainManager API.			
AdministratorV1			Show/Hide	List Operations	Expand Operations
GET	/api/v1/notifications				
GET	/api/v1/mmmlink				

AssetsV1

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/v1/tenancies	The operation gets multiple tenancies from the MainManager API.
POST	/api/v1/tenancies	The operation creates tenant with the MainManager API.
PUT	/api/v1/tenancies	The operation creates or update tenant with the MainManager API.
GET	/api/v1/sites	The operation gets multiple sites from the MainManager API.
POST	/api/v1/sites	The operation creates sites with the MainManager API.
PUT	/api/v1/sites	The operation creates or update sites with the MainManager API.

Assets

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/restapi/Assets/GetSites	The web service will try to find all sites that fit the given search criteria.
GET	/restapi/Assets/GetSitesFull	The web service will try to find all sites that fit the given search criteria.
GET	/restapi/Assets/GetBuildings	Find all active buildings that belong to a specific site identification
GET	/restapi/Assets/GetBuildingsFull	Find all active buildings that belong to a specific site identification
GET	/restapi/Assets/GetFloors	The web service will try to find all floors that fit the given search criteria.
GET	/restapi/Assets/GetFloorsFull	The web service will try to find all floors that fit the given search criteria.
GET	/restapi/Assets/GetSpaces	The web service will try to find all spaces that fit the given search criteria.
GET	/restapi/Assets/GetSpacesFull	The web service will try to find all spaces that fit the given search criteria.

IncidentV1

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/v1/incidents	
POST	/api/v1/incidents	
PUT	/api/v1/incidents	
GET	/api/v1/incidentcategorieslookup	The web service will try to find all check words that fit the given search criteria.
GET	/api/v1/incidentsubcategorieslookup	The web service will try to find all check word items that fit the given search criteria.
GET	/api/v1/incidentslookup	

OrganisationV1

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/v1/vendors	The operation gets multiple vendors from the MainManager API.
POST	/api/v1/vendors	The operation creates vendors with the MainManager API.
PUT	/api/v1/vendors	The operation creates or update vendor with the MainManager API.

PropertyManagementV1

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/v1/leasecontracts	The operation gets multiple lease contracts from the MainManager API.
POST	/api/v1/leasecontracts	The operation creates lease contract with the MainManager API.
PUT	/api/v1/leasecontracts	The operation creates or update lease contract with the MainManager API.
GET	/api/v1/leasecontracttasks	
POST	/api/v1/leasecontracttasks	
PUT	/api/v1/leasecontracttasks	

Authentication

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/restapi/token	Authenticates provided credentials and returns an access token
------	----------------	--

As part of the REST API we provide subscription services (webhooks) for external systems to receive add/change/delete updates from the system. Then the subscriber must implement one or more endpoint of the following types:

AssetsChangeEvent

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/restapi/AssetsChangeEvent/SiteChange	This service receives a change event on a site from external system from which this system subscribes to.
POST	/restapi/AssetsChangeEvent/BuildingChange	This service receives a change event on a building from external system from which this system subscribes to.
POST	/restapi/AssetsChangeEvent/FloorChange	This service receives a change event on a building floor from external system from which this system subscribes to.
POST	/restapi/AssetsChangeEvent/SpaceChange	This service receives a change event on a building space from external system from which this system subscribes to.

The definition is accessible in the standard swagger documentation.

5.1.2. Asset information SOAP API

Following methods are accessible:

- [GetAssetList](#)
- [GetBuildingAddressList](#)
- [GetBuildingFloors](#)
- [GetBuildingList](#)
- [GetBuildingSpaces](#)
- [GetBuildings](#)
- [GetEmployeeList](#)
- [GetGroundAddressList](#)
- [GetGroundList](#)
- [GetOpenAreaAddressList](#)
- [GetOpenAreaList](#)
- [GetPropertyValuationBuildingList](#)
- [GetRealPropertyAddressList](#)
- [GetRealPropertyList](#)
- [GetRegionList](#)

See detailed information in respective API documentation.

5.1.3. Main processes SOAP API

Following methods are accessible:

- [AcceptedHourExportResult](#)
- [AcceptedMaterialExportResult](#)
- [CreateAccount](#)
- [CreateAccount2](#)
- [CreateAccountKey](#)
- [CreateConditionValue](#)
- [CreateEmployee](#)
- [CreateIncident](#)
- [CreateProjectSection](#)
- [CreatePurpose](#)
- [CreateTimeRegistration](#)
- [CreateTransaction](#)
- [CreateTransaction2](#)
- [GetAcceptedHours](#)
- [GetAcceptedMaterial](#)
- [GetCustomerInfo](#)
- [GetRequests](#)
- [InsertFromFocal](#)
- [RequestExportResult](#)
- [TransactionsImportFinished](#)
- [UpdateRequest](#)

See detailed information in respective API documentation.

5.2. Asset data integrations

5.2.1. BBR in Denmark

5.2.2. Matrikkel in Norway

5.2.3. Fasteignamat in Iceland

5.3. Migration methods for Assets

Multiple migration actions have been implemented in the system. Following sections lists those actions.

5.3.1. Import Structure\Sites (Ground)

Name	Description	Prerequisite
ImportBuildingTreeFromClipboard	<i>This action method can be found under sites</i> Creates main group and ground. Creates new buildings, floors, space and space part. The only mandatory attribute are site name and site nr.	None
ImportBuildingTreeFromClipboard2	<i>This action method can be found under sites</i> Creates main group and ground. Creates new buildings, floors, space and space part. Mandatory attribute are site name, site nr and space.	None
ImportGroundFromClipboard1	<i>This action method can be found under sites</i> Import grounds from clipboard. The only mandatory attribute are site name, site nr and region nr.	Region must exist Import Region

5.3.2. Import Structure\Region

Name	Description	Prerequisite
ImportRegionFromClipboard	<i>This action method can be found under sites</i> Import regions from clipboard. The only mandatory attribute are name and nr.	None

5.3.3. Import Structure\Real Property

Name	Description	Prerequisite
ImportRealPropertyFromClipboard1	<i>This action method can be found under sites</i> Import real properties from clipboard 1. Mandatory attributes are ground nr, land id and name. This method creates new city, Owning condition and type if it does not exists.	Ground must exist. Import Sites (Ground)

5.3.4. Import Structure\Buildings

Name	Description	Prerequisite
ImportBuildingFromClipboard1	<i>This action method can be found under buildings.</i> Imports buildings from clipboard where ground exists and real property.	Real Property and Ground must exist. Import Real Property

	Creates Property value bindings.	Import Sites (Ground)
--	----------------------------------	---------------------------------------

5.3.5. Import Structure\Building Space

Name	Description	Prerequisite
ImportApartmentFromClipboard	<i>This action method can be found under sites and Service Contract.</i> Creates new apartments on existing ground, building, floor and space. Mandatory attributes Ground locator, building, floor, apartment, apartment locator.	Ground, Building, Floor, and Space must exist. Import Sites (Ground) Note this is the only method available that creates floors.
ImportBuildingSpaceFromClipboard1	<i>This action method can be found under Building Space or Service Contract(Lease)</i> Creates a new floor on existing ground, building, apartment and space. Mandatory attributes Ground locator, Building locator, Apartment locator.	Ground, Building, Apartment must exist. Import Sites (Ground)

5.4. Migration methods for Organizations and Human resources

5.4.1. Import Organizations\Company

Name	Description	Prerequisite
ImportCompanyFromClipboard1	<i>This action method can be found under Organizations/organizations.</i> Imports Companies, Departments, Divisions, Branches and Branch sections from clipboard. Creates or updates if used. Mandatory attributes Company identification, name, .relationship, address and status.	Relationship ID must be one of: 1 [Own company] 2 [Customer] 3 [Partner] 4 [Supplier]
ImportOrganisationFromClipboard1	<i>This action method can be found under Organizations Employees.</i> Import organization structure from clipboard. . Mandatory attributes Company identification and name	Parent organization must exist if used
ImportCompanyDataFromClipboard	<i>This action method can be found under Organizations\organizations.</i> Import company data from clipboard. FindsOrCreates Organization and employee. Connects to organizational	Account and Ground must exist Import Structure Sites (Ground) Import

	unit if used.	Accounts\Accounts
ImportOrgAndEmpFromClip1	Afsdaf This action makes is possible to import organizations and employees in the same clip board action. To understand the parameter please read OrganisationAPIDesign.docx Clip board can be imported in 3 different ways. 1. Only organizations 2. Organizations and employees together Only employees	

5.4.2. Import Organizations\Employees

Name	Description	Prerequisite
ImportEmployeeFromClipboard1	<i>This action method can be found under Organizations Employees.</i> Import employees from clipboard. This import will set the Origin of registration from Agresso. Creates or updates if used. Mandatory attributes Company identification, name and employee resource number	none

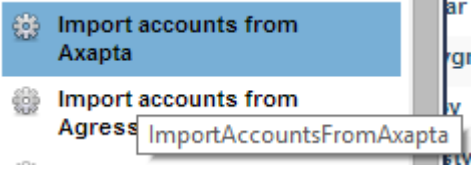
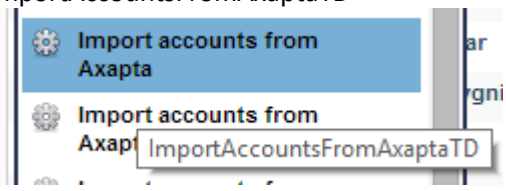
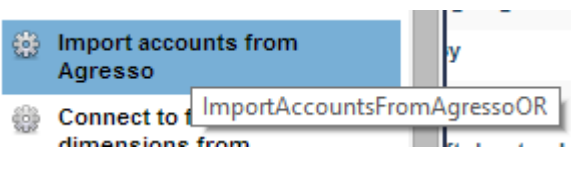
5.5. Import methods for Plans

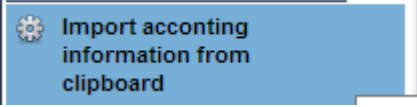
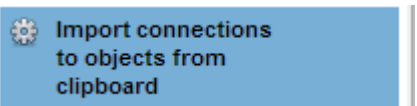
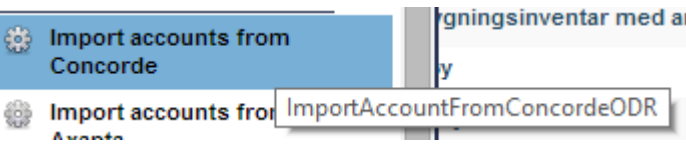
Name	Description	Prerequisite
ImportPlanFromClipboard1 acImportFromClipboard	<i>This action can be found under Task planning.</i> Import action that import yearly plan from the clipboard (Excel).	

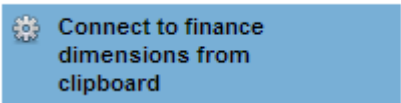
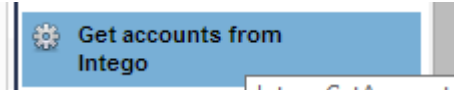
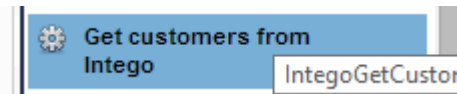
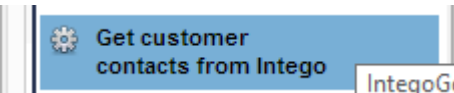
5.6. Migration and integration methods for Finance management

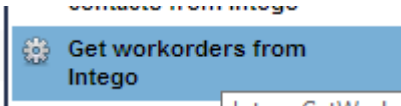
5.6.1. Import Accounts\Accounts

Name	Description	Prerequisite
ImportAccountsFromAxapta	<i>This action method can be found under Finance and Accounts.</i> This is an automatic import action that integrates data	Foreign session to Axapta database need to be available.

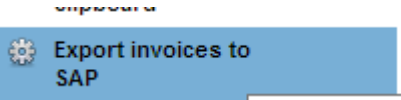
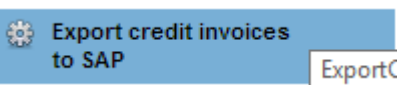
 acImportFromOtherSystem	from Axapta and inserts into MainManager. This action is customer specific and not general. This import creates Account, Project Section and Project Section type.	MainManger windows service needs to be running.
 acImportFromOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic import action that integrates data from Axapta and inserts into MainManager. This action is customer specific and not general but it can be adapted by having same source table structure. This import creates Account, AccountForMainObject, Account Group, Project Group, Project Section and Project Section type, Project Activity, Project Activity Type and Account Type.	Foreign session to Axapta database needs to be available. MainManger windows service needs to be running.
 acImportFromOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic import action that integrates data from Agresso and inserts into MainManager. This action is customer specific and not general but it can be adapted by having same source table structure. This import creates Account, Project Section and Project Section type	Foreign session to Agresso database needs to be available. MainManger windows service needs to be running.
aImportAccountingInfoFromClipboard	<i>This action method can be found under Finance</i>	FlexValueSetName must by one of following:

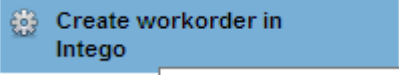
 acImportFromClipboard	and Accounts. This is a non-automatic clipboard import action that inserts data into MainManager. This action is customer specific and not general but it can be adapted by having same source table structure. This import creates Account, Cost Center and Section Post.	XXSB_KONTO , XXSB_KSTED or XXSB_KAPPOST
 acImportFromClipboard	This action method can be found under Finance and Accounts. This is a non-automatic clipboard import action that inserts data into MainManager. This action is general. This import creates AccountForMainObject that connects mainObject with Account.	Ground and Main ID must exist in MainManager. Account Nr must exist in MainManager.
aImportAccountFromConcordeODR  acImportFromOtherSystem	This action method can be found under Finance and Accounts. This is an automatic import action that integrates data from Agresso and inserts into MainManager. This action is customer specific and not general but it can be adapted by having same source table structure. This import creates Account, Account Type, Project Section, and Project Section type, Project Activity, Project Activity Type and Person.	Foreign session to Concorde database needs to be available. MainManger windows service needs to be running.
ConnectToFinanceDimensionsFromClipboard	This action method can be found under Finance	Account must exist. Division must exist. Cost

Agresso  acImportFromClipboard	and Accounts. This is a non-automatic clipboard import action that inserts data into MainManager. This action is general. This import updates existing account.	center must exist. Account key must exist. Purpose must exist.
IntegoGetAccounts  acImportFromOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic import action that integrates data from Intego and inserts into MainManager. This action is customer specific and not general but it can be adapted by having web service structure. This import creates Account. Name and Nr.	Foreign session to web service needs to be available. MainManger windows service needs to be running.
IntegoGetCustomers  acImportFromOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic import action that integrates data from Intego and inserts into MainManager. This action is customer specific and not general but it can be adapted by having web service structure. This import creates Ground, Organization and adds account to ground.	Foreign session to web service needs to be available. MainManger windows service needs to be running.
IntegoGetCustomerContacts  acImportFromOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic import action that integrates data from Intego and inserts into MainManager. This action is customer specific and not general but it can be adapted by	Foreign session to web service needs to be available. MainManger windows service needs to be running.

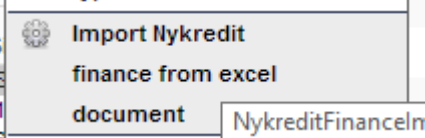
	having web service structure. This import creates Employee.	
IntegoGetWorkorders  acImportFromOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic import action that fetches work orders from Intego and displays the data. This action is customer specific and not general but it can be adapted by having web service structure. This action does not import any data.	Foreign session to web service needs to be available. MainManger windows service needs to be running.

5.6.2. Export Accounts\Accounts

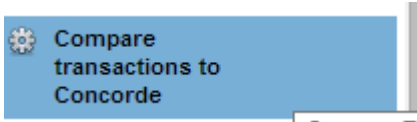
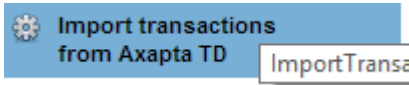
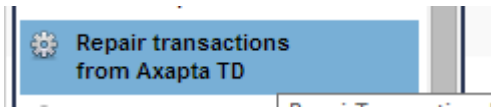
Name	Description	Prerequisite
aExportInvoicesToSAP  acExportToOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic export action that integrates data from MainManager and inserts into SAP. This action is customer specific and not general but it can be adapted by having same destination table structure. This export action exports Invoices and writes the result into a text file.	Foreign session to export path needs to be available. MainManger windows service needs to be running.
aExportCreditInvoicesToSAP  acExportToOtherSystem	<i>This action method can be found under Finance and Accounts.</i> This is an automatic export action that integrates data from MainManager and inserts into SAP. This action is customer specific and not general but it can be adapted by having same destination table structure. This export action exports Invoices and writes the result into a text file.	Foreign session to export path needs to be available. MainManger windows service needs to be running.
IntegoCreateWorkOrder	<i>This action method can be found under Finance and Accounts.</i> This	Foreign session to web service path needs to

 acExportToOtherSystem	is an automatic export action that integrates data from MainManager and inserts into SAP. This action is customer specific and not general but it can be adapted by having same destination wsdl. This export action creates or updates a work order via web service.	be available.

5.6.3. Import Finance\Transactions

Name	Description	Prerequisite
NykreditFinanceImport  acImportFromOtherSystem	<i>This action method can be found under Finance and Transactions.</i> This is a non automatic import action that integrates data from excel file and inserts into MainManager. This action is general. This import updates selected transaction.	Path to excel file must exist. Excel sheet must follow given standard.
aImportTransactionsFromOracleBusinessSuiteFR  acImportFromOtherSystem	<i>This action method can be found under Finance and Transactions.</i> This is an automatic import action that integrates data from ftp and inserts into MainManager. This action is customer specific and not general. This import action reads file from FTP and creates transactions.	Path to ftp must exist. MainManger windows service needs to be running.
ImportTransactionsFromConcordeODR	<i>This action method can</i>	Foreign session to

 Import transactions from Concorde acImportFromOtherSystem	<i>be found under Finance and Transactions.</i> This is an automatic import action that integrates data from Concorde and inserts into MainManager. This action is customer specific and not general, but with same source table structure it can be reused. This import action reads data from Concorde database and creates transactions.	Concorde database needs to be available. MainManger windows service needs to be running.
ReimportTransactionsFromConcordeODR  Reimport transactions from Concorde acImportFromOtherSystem	<i>This action method can be found under Finance and Transactions.</i> This is a non-automatic import action that integrates data from Concorde and inserts into MainManager. This action is customer specific and not general, but with same source table structure it can be reused. This import action reads data from Concorde database and creates transactions.	Foreign session to Concorde database needs to be available. MainManger windows service needs to be running.
ImportTransactionDocumentsFromConcordeODR  Import transaction documents from Concorde acImportFromOtherSystem	<i>This action method can be found under Finance and Transactions.</i> This is an automatic import action that integrates data from Concorde and inserts into MainManager. This action is customer specific and not general, but with same source table structure it can be reused. This import action imports	Foreign session to Concorde database needs to be available. And document path must exist. MainManger windows service needs to be running.

	documents from Concorde database. Updates transactions with documents.	
CompareTransactionsToConcordeODR  acImportFromOtherSystem	<i>This action method can be found under Finance and Transactions.</i> This is an automatic import action that counts transactions in Concorde and MainManager. This action is customer specific and not general, but with same source table structure it can be reused. This action counts rows and returns error if mismatch.	Foreign session to Concorde database needs to be available. And document path must exist. MainManger windows service needs to be running.
ImportTransactionsFromAxaptaTD  Or RepairTransactionsFromAxaptaTD 	<i>This action method can be found under Finance and Transactions.</i> This is an automatic import action that integrates data from Axapta and inserts into MainManager. This action is customer specific and not general, but with same source table structure it can be reused. This import action reads data from Axapta database and creates Vendors, Purpose, Accounts, Account Key, cost center, cost type. This action can also be used to repair transactions.	Foreign session to Axapta database needs to be available. MainManger windows service needs to be running.

5.6.4. Import Contracts\Lease Contracts

Name	Description	Prerequisite
ImportObjectAndLeaseContractFromClipboard	<i>This action method can be found under Contract and Lease</i>	Ground, Building, Apartment must exist.

	contracts. Imports building structure and lease contracts from clipboard. Mandatory attributes Ground locator, Building, Floor, Apartment and Contract. This import finds or creates organizations.	Import Sites (Ground) Import apartment
ImportLeaseContractDataFromClipboard1	<i>This action method can be found under Contract and Lease contracts.</i> Import lease contracts from clipboard. Mandatory attributes apartment location and Payer.	Ground, Building, Apartment must exist. Import Sites (Ground) Import apartment
ImportLeaseContractFromClipboard	<i>This action method can be found under Contract and Lease contracts.</i> Import lease contracts from clipboard. Mandatory attributes ground, contract name, and contract nr and netto area.	Ground must exist. Building and real property must exist if used. Import Structure Sites (Ground) If used Import Structure Real Property Import Structure Buildings
ImportLeaseContractHolderFromClipboard	<i>This action method can be found under Contract and Lease contracts.</i> Import lease contract holders from clipboard and creates companies. Mandatory attributes holder name and identification.	Zip code must exist if used.

5.7. Integration methods for Operation management

TBD.

5.8. Integration methods for Service management

TBD.

5.9. Integration methods for Administration

TBD.

5.10. Integration API - Uncategorized

Name	Description
<u>GetTowers</u>	This method returns a list of transmission line towers. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/MainObjects.asmx?op=GetTowers
<u>GetBuildingFloors</u>	This method returns an instance of <u>BuildingFloor</u> . Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/MainObjects.asmx?op=GetBuildingFloors
<u>GetBuildingSpaces</u>	This method returns an instance of <u>BuildingSpace</u> . Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/MainObjects.asmx?op=GetBuildingSpaces
<u>GetBuildings</u>	This method returns an instance of <u>Building</u> . Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/MainObjects.asmx?op=GetBuildings
<u>GetStartEquipment</u>	This method returns a list of <u>Equipment</u> . Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/MainObjects.asmx?op=GetStartEquipment
<u>GetStartZone</u>	This method returns a list of <u>Zone</u> . Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ MainProcesses.asmx?op=GetStartZone
<u>AcceptedHourExportResult</u>	This method returns result on export on accepted hours from Excel. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ MainProcesses.asmx?op=AcceptedHourExportResult The result contains success variable that is true if the operation was successful.
<u>AcceptedMaterialExportResult</u>	This method returns result on export on accepted material from Excel. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ MainProcesses.asmx?op=AcceptedMaterialExportResult

	The result contains success variable that is true if the operation was successful.
<u>GetAcceptedHours</u>	This method returns a list of <u>xAcceptedHours</u> .. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ MainProcesses.aspx?op=GetAcceptedHours
<u>GetAcceptedMaterial</u>	This method returns a list of <u>xAcceptedMaterial</u> .. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ MainProcesses.aspx?op=GetAcceptedMaterial
<u>GetCustomerInfo</u>	This method returns an instance of <u>CustomerInfo</u> for a given customer id. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ MainProcesses.aspx?op=GetCustomerInfo
<u>GetRequests</u>	This method returns a list of <u>xRequest</u> .. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ MainProcesses.aspx?op=GetRequests
<u>RequestExportResult</u>	This method returns result on excel export result. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/MainProcesses.aspx?op=RequestExportResult The result contains success variable that is true if the operation was successful.
<u>GetGaugeData</u>	This method returns a list of <u>xGauge</u> .. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ EnergyProvider.aspx?op=GetGaugeData
<u>GetGaugeLocations</u>	This method returns an instance of <u>WebFunctions</u> . Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ EnergyProvider.aspx?op=GetGaugeLocations
<u>GetGauges</u>	This method returns an list of <u>xListItem</u> . Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ EnergyProvider.aspx?op=GetGauges
<u>GetMeters</u>	This method returns a list of <u>xGauge</u> .. Main manager has this web service available. This method is part of the main object service layer that is use for exporting data from Main manager. The method can be found at: <Webserver>/Services/ EnergyServices.aspx?op=GetMeters
<u>AppAuthenticate</u>	Web service for the app. Not a service layer. <Webserver>/Services/AppProcesses.aspx
<u>AppAuthenticateExten</u>	Web service for the app. Not a service layer. <Webserver>

<u>ded</u>	/Services/AppProcesses.asmx
<u>AppAuthenticateUser</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>AppGetMMDataset2</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>AppGetMMSystemSet2</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>AppGetMyIncidentDataSet</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>AppGetRequests</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>AppGetSiteFromAppSiteKey</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>AppGetUserType</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>GetMainDbUpdates</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>GetMainModuleAccess</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>GetMainModuleGroupAccess</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>GetMainModuleItemAccess</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>GetNationalRegisterData</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>GetOrAddTranslation</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>GetTranslations</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>GetUserRoles</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>UserName</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx

Import web service	
Name	Description
<u>CreateAccount</u>	This method creates or updates account by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver> /Services/MainProcesses.asmx?op=CreateAccount The result contains success variable that is true if the operation was successful.
<u>CreateAccount2</u>	This method creates or updates account key by given parameters and stores in database. Main manager has this web service available. This method is

	part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreateAccount2 The result contains success variable that is true if the operation was successful.
<u>CreateAccountKey</u>	This method creates or updates account by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreateAccountKey The result contains success variable that is true if the operation was successful.
<u>CreateEmployee</u>	This method creates or updates employee by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreateEmployee The result contains success variable that is true if the operation was successful.
<u>CreateIncident</u>	This method creates or updates incident by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreateIncident The result contains success variable that is true if the operation was successful.
<u>CreateProjectSection</u>	This method creates or updates project section by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreateProjectSection The result contains success variable that is true if the operation was successful.
<u>CreatePurpose</u>	This method creates purpose project section by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreatePurpose The result contains success variable that is true if the operation was successful.
<u>CreateTimeRegistration</u>	This method creates or updates time registration by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreateTimeRegistration The result contains success variable that is true if the operation was successful.
<u>CreateTransaction</u>	This method creates or updates time transaction by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver>/Services/MainProcesses.asmx?op=CreateTransaction

	The result contains success variable that is true if the operation was successful.
<u>CreateTransaction2</u>	This method creates or updates time transaction by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver> /Services/MainProcesses.asmx?op=CreateTransaction2 The result contains success variable that is true if the operation was successful.
<u>InsertFromFocal</u>	This method creates an incident service request by given parameters and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver> /Services/MainProcesses.asmx?op=InsertFromFocal The result contains success variable that is true if the operation was successful.
<u>TransactionsImportFinished</u>	This method corrects actual cost for active transactions. Updates transactions in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver> /Services/MainProcesses.asmx?op=TransactionsImportFinished The result contains success variable that is true if the operation was successful.
<u>UpdateRequest</u>	This method Updates requests and stores in database. Main manager has this web service available. This method is part of the main object service layer that is use for importing data into Main manager. The method can be found at: <Webserver> /Services/MainProcesses.asmx?op=UpdateRequest The result contains success variable that is true if the operation was successful.
<u>AppPushMMDataset2</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>SaveImage</u>	Web service for the app. Not a service layer. <Webserver> /Services/AppProcesses.asmx
<u>AddAppSiteKey</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>AddAppUser</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>AddEditMMDatabase</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>AddEditMMWebsite</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx
<u>AddMainDbUpdate</u>	Web service for the app or Mainmanger. Not a service layer. <Webserver> /Services/ UserSettings.asmx

Agresso	
Name	Description
<u>ImportServiceContractFromAgressoOR</u>	This method creates service contracts in MainManager with Aggresso data. This method connects to Aggresso database, fetches service

	contracts and imports Aggresso data into MainManager. This method is not dependent on user input data.
<u>UpdateEmployeeFromAgresso</u>	This method fetches employee data from Aggresso via web service and imports into MainManager. This method can be started on demand from MainManager. When importing: MainManager creates or updates an employee and connects the employee to a correct organization. A difference report is created after this import.
<u>UpdateOrganisationFromAgresso</u>	This method fetches organization data from Aggresso via web service and imports into MainManager. This method can be started on demand from MainManager. When importing: MainManager creates or updates an organization.

AutoCAD	
Name	Description
<u>AutoCADImageExportTest</u>	Image export test from AutoCAD model
<u>ImportAutocadFacilityGIS</u>	Imports AutoCAD drawing to Facility GIS
<u>ImportAutocadFacilityGISStatsbygg</u>	Imports objects in the AutoCAD drawing and makes an interactive drawing
<u>ImportAutocadFacilityGISStatsbygg</u>	Imports objects in the AutoCAD drawing and makes an interactive drawing
<u>ImportAutocadInteractiveDrawingHedmark</u>	Imports objects in the AutoCAD drawing and makes an interactive drawing
<u>ImportEquipmentAndMetersFromAutoCAD Drawing</u>	Imports equipment and meters from AutoCAD drawings.
<u>ShowImportResultForObject</u>	Show import results from import of objects from AutoCAD drawing
<u>ShowImportResultForObject</u>	Show import results from import of objects from AutoCAD drawing
<u>ShowImportResultForObject</u>	Show import results from import of objects from AutoCAD drawing

Axapta	
Name	Description
<u>ImportCustomerFromAxaptaTD</u>	Import customers from Axapta

Bakvordur	
Name	Description
<u>UpdateEmployeeFromBakvordur</u>	This method fetches employee data for one given Employee number from Bakvordur via web service and imports new data into MainManager. This method can be started on demand from MainManager. When importing: MainManager creates or updates an employee and connects the employee to a correct organization and saves employee picture into MainManager.
<u>UpdateOrganisationFromBakvordur</u>	This method fetches all organizations from

	Bakvordur via web service and imports new data into MainManager. This method can be started on demand from MainManager. When importing: MainManager updates all employees and connects the employee to a correct organization and saves employee picture into MainManager.
	Update organization from Bakvörður

BIMSYNC	
Name	Description
BIMSYNCImportElement	Import BIMSYNC elements
BIMSYNCImportModel	Import BIMSYNC models
BIMSYNCImportProject	Import BIMSYNC projects

CareTaker	
Name	Description
CaretakerFinanceImport	Import text file from CareTaker

Concorde	
Name	Description
CompareRequestToConcordeODR	Check if requests already exist in Concorde and mark them accordingly
ImportUnitpriceTypeFromConcordeODR	Import stock types from Concorde
ReimportUnitpriceTypeFromConcordeODR	Reimport stock types from Concorde
ImportDeliveriesFromConcordeODR	Import deliveries from Concorde
InactivateDeliveriesThatDoNotExistInConcordeODR	Inactivate deliveries that do not exist in Concorde
ImportPayZoneFromConcorde	Import pay zones from Concorde
ImportStockTypeFromConcordeODR	Import stock types from Concorde
ReimportStockTypeFromConcordeODR	Reimport stock types from Concorde
ImportStockTypeFromConcordeODR	Import stock types from Concorde
ReimportStockTypeFromConcordeODR	Reimport stock types from Concorde
ImportStockTypeFromConcordeODR	Import stock types from Concorde
ReimportStockTypeFromConcordeODR	Reimport stock types from Concorde
ImportCustomerFromConcordeODR	Import customers from Concorde
ImportStockTypeFromConcordeODR	Import stock types from Concorde
ReimportStockTypeFromConcordeODR	Reimport stock types from Concorde

CoreFM	
Name	Description

<u>XMLImportActivitiesFromCoreFM</u>	Imports activities from CoreFM system in xml format
<u>XMLImportCompaniesFromCoreFM</u>	Imports companies from CoreFM system in xml format
<u>XMLImportLocationFromCoreFM</u>	Imports location from CoreFM system in xml format
<u>XMLImportObjectsFromCoreFM</u>	Imports objects from CoreFM system in xml format

Rambygg	
Name	Description
<u>ImportFromRambygg</u>	Import from Rambygg

IFC	
Name	Description
<u>ImportLocationFromIFC</u>	Imports location structure from a IFC file
<u>ImportIFCInformation</u>	Imports information from the selected IFC document

Intergraph	
Name	Description
<u>ImportIncidentsFromIntergraph(Appointment)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Environmental task)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Failure)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Fault)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Incident)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Incident/ fault report)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Maintenance incident)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Purchase Order)</u>	Imports incidents from Intergraph
<u>ImportIncidentsFromIntergraph(Release Item)</u>	Imports incidents from Intergraph

LeifLarsen	
Name	Description
<u>ImportLeifLarsenAccess</u>	Imports data from Leif Larsen Access database

OIOREST	
Name	Description
<u>ImportAddressesFromOIOREST</u>	Imports addresses from OIOREST
<u>ImportCountryStatesFromOIOREST</u>	Imports country states from OIOREST
<u>ImportRoadsFromOIOREST</u>	Imports roads from OIOREST
<u>ImportZipcodesFromOIOREST</u>	Imports zip codes from OIOREST

OracleBusinessSuiteFR	
Name	Description
<u>ImportTransactionsFromOracleBusinessSuiteFR</u>	Import transactions from Oracle Business Suite for FR, through FTP

PropMan	
Name	Description
<u>ImportFromPropMan</u>	Import regions, grounds, real properties and buildings from PropMan
<u>ImportFromPropManService</u>	Import Service Contacts and Service types from PropMan
<u>ImportLeaseDataFromPropMan</u>	Import lease contracts and lease holders from PropMan

CreateIncident : When incidents are created in MainManager some Customers have the need that it is created in another system

GetIncidents: When incidents are created in MainManager some Customers have the need that it is created in another system. This function returns incidents

GetGrounds: Returns grounds with defined attributes

GetBuildings: Returns buildings with defined attributes

GetBuildingFloors: Returns building floors with defined attributes

GetBuildingSpaces: Returns building spaces with defined attributes

GetTransmissionLines: Returns transmission lines/towers with defined attributes

GetHVEquipment: Returns HV equipment with defined attributes

ComputerScan: Scans network and registers computers into MainManager

ImportSoftware: Scans computers for software and registers into MainManager

ImportFromAD: Imports/updates users from Active directory

ImportFromFMR: Imports/updates grounds/buildings/spaces and estates from the Icelandic building register

ImportFromNationalRegister: Queries the Icelandic national register for import into MainManager

5.11. Integration with GIS systems

General

Generel process som vi går igennem inden vi udfører forbindelser mellem MainManager og GIS system er efterfølgende:

1. Beslutning om hvad skal bruges til at forbinde registreringer i MainManager og registreringer i GIS system sammen. Denne forbindelse må være en form for en unik nøgle. F.eks. har vi brugt matrikel nummer for ejendomme/matrikler hos Reykjavik kommune (kunde i Island) og så har vi brugt ID nummer fra MainManager databasen til at forbinde til sammen med transmissionslinjer for Landsnet (en kunde i Island).

2. Vi skal undersøge hvilken teknologi GIS systemet tilbyder omkring visning. Vi har brugt Flash (hos Samsýn, GIS software leverandør baseret i Island), vi har brugt JQuery funktioner (hos Loftmyndir, GIS data leverandør og kort udgiver i Island og Intergraph, global leverandør af GIS software).
3. Finde ud af hvilke muligheder GIS systemet har til at kommunikere tilbage til et andet system (som MainManager). Vi har udført det således at MainManager definerer JQuery/JavaScript funktioner som GIS systemet kalder på når en bruger åbner kortet og bevæger sig i det. F.eks. vælger brugeren en matrikel i kortet, bliver en funktion kaldt op som indeholder unik nøgle for matrikel som dermed vælger den samme matrikel / ejendom i MainManager systemet.

Flash

GIS er vist som en Flash fil (*.swf) i MainManager. Ved styring af GIS (størrelse, zoomlevel) sender MainManager parametre ved brug af "http querystring".

Eksempel af et GIS Flash forbindelse(Samsýn - Reykjavik):

```
<object                                classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0"
width="1344"      height="300"      id="Gatnavidhald"      align="lt"><param      name="movie"
value="http://rvk.mainmanager.is:80/Scripts/LUKR/stjorinnMap.swf?landnr=111278&loftmynd=true">
<param name="quality" value="high"><param name="scale" value="noscale"><param name="salig"
value="lt"><param      name="bgcolor"      value="#CCCCCC"><param      name="wmode"
value="transparent"><embed                                wmode="transparent"
src="http://rvk.mainmanager.is:80/Scripts/LUKR/stjorinnMap.swf?landnr=111278&loftmynd=true"
quality="high"  bgcolor="#ffffff"  width="1344"  height="300"  name="MapReykjavik"  align="lt"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer"></object>
```

Som et eksempel så vil efterfølgende http kald vise et billede af gaden „Laugaveg 32“ i størrelsen 430*420.

<http://mapserver.is/website/map.swf?gotuheiti=Laugavegur&husnumer=32h=430&w=420&loftmynd=true>

GIS kommunikerer med MainManager med brug af Javascript funktioner. For eksempel tager efterfølgende funktion i mod parametre om lokation når et „click“ sker på GIS kortet:

```
function sendPointMap(mapurl,pointx,pointy,zoomlevel)
{
    // MainManager modtager parametre og arbejder videre med dem
}
```

JQuery

En reference til Javascript API er lagt ind i MainManager(Intergraph):

```
<script type="text/javascript" src="http://mainmanager.imaps.nu/map.js"></script>
```

Intergraphs Javascript API bruger et „<DIV>“ objekt til visning af GIS kort.

Eksempel:

```
<script language="javascript" type="text/javascript">
$m.init('map', , ,function ()),function(lat,lon){ MarkPoint(lat, lon);});

</script>
<div id="map">
</div>
```