Advanced Applications Module (workflow configuration)

Contents

Introduction

The WAY4™ Advanced Applications R2 module is used to create new system objects (clients, contracts, etc.) and modify parameters of existing objects. It does so by processing manually entered applications, as well as those imported from files or online.

This document describes the configurations required for operation of version R2 of the Advanced Applications module.

The following conventions are used in the document:

- Field names in screen forms are displayed in *italics*.
- Button labels in screen forms are encased in square brackets, as in [Approve].
- Sequences for selection of user menu items are shown using arrows, as in the following example: "Full → Issuing → Contracts Input & Update".
- Key combinations used when working with DB Manager are encased in angular brackets, for example <Ctrl>+<F3>.
- Warnings about actually hazardous situations or actions are marked with the ⚠ sign.
- Messages marked with the ℹ sign contain information about important features, additional options, or the best use of certain system functions.

Chapter 1. Advanced Applications Module Purpose and Operating Principle

Applications are processed according to a selected workflow sequence (strategy).

A strategy is an earlier configured sequence of workflow stages.

Within the strategy, earlier configured workflow activities are registered, as well as rules. Rules are a set of conditions and workflow activities when these conditions are met.

Strategy components are workflow stages (steps) that can also be characterized by a set of activities and rules used within the step.

Within a strategy, steps are registered in a specific sequence, corresponding to the workflow algorithm for a particular application. For example, the sequence of operations to process an application for issuing a client debit bankcard includes entry of data to register the client and card contract, verifying the entered data and the operation to issue the card itself. When processing an application to issue a credit bankcard, a client scoring procedure is a required workflow step. A workflow algorithm is determined by bank regulations.

For the strategy as a whole, as well as for separate steps, rules are configured ensuring that when the set conditions are met, the application is sent to another step of the strategy or to the first step of another strategy.

After registering an application as a result of manual input or import from a file, the system selects the workflow strategy. Selection is made according to the application's parameter values.

Applications are processed without direct user access to database system tables (clients, contracts and documents). During application processing, all changes to application parameters are reflected only in module tables.

When an application has been processed, a special Accept procedure transfers data from the Advanced Applications module directly into tables of other system modules for registration of contracts, clients, etc.

When an application has been processed, it is given the Posted status, after which the application is filtered out of the list of applications for processing.

Applications created using screen forms are built according to a template, called an application scheme.

An application scheme makes it possible to set a main object (client or contract) for which the application is intended, and a hierarchy of related objects.

A scheme consists of sections, contains a main section (to enter data about the main object) and a set of subsections organized in a hierarchical structure. Each section is used to enter information about a system object (clients, contract, address, etc.).

Example of hierarchical structure of objects:

- Client record (main object).
  - Contract.
    - Address.
    - Card contract.

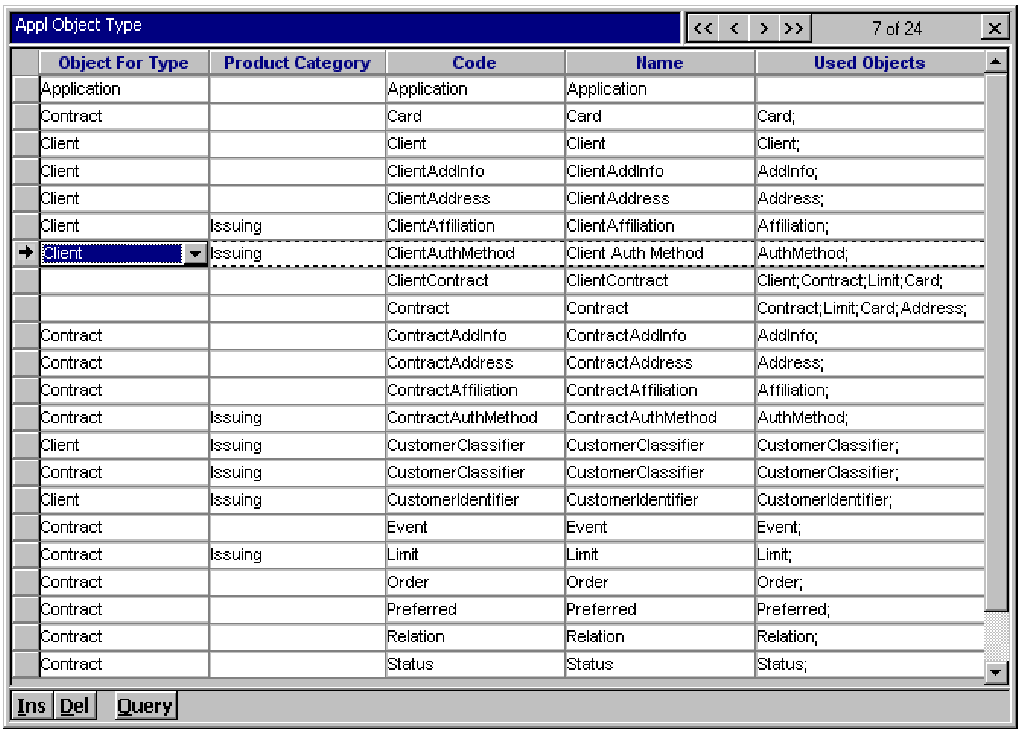
Chapter 2. Service Catalogues

Common catalogues are used to set lists of objects (Advanced Applications R2 → Setup → Appl Object Type) for which applications, permitted actions, etc. are intended.

Service catalogues include: the catalogue of application objects, catalogue of actions with objects, application additional information classifiers and Product codes.

Catalogue of Objects

The catalogue of objects (Advanced Applications R2 → Setup → Appl Object Type) contains a list of system objects with which the Advanced Applications module can operate (see Fig. 1).



Object For Type	Product Category	Code	Name	Used Objects
Application		Application	Application	
Contract		Card	Card	Card;
Client		Client	Client	Client;
Client		ClientAddInfo	ClientAddInfo	AddInfo;
Client		ClientAddress	ClientAddress	Address;
Client	Issuing	ClientAffiliation	ClientAffiliation	Affiliation;
Client	Issuing	ClientAuthMethod	Client Auth Method	AuthMethod;
		ClientContract	ClientContract	Client;Contract;Limit;Card;
		Contract	Contract	Contract;Limit;Card;Address;
Contract		ContractAddInfo	ContractAddInfo	AddInfo;
Contract		ContractAddress	ContractAddress	Address;
Contract		ContractAffiliation	ContractAffiliation	Affiliation;
Contract	Issuing	ContractAuthMethod	ContractAuthMethod	AuthMethod;
Client	Issuing	CustomerClassifier	CustomerClassifier	CustomerClassifier;
Contract	Issuing	CustomerClassifier	CustomerClassifier	CustomerClassifier;
Client	Issuing	CustomerIdentifier	CustomerIdentifier	CustomerIdentifier;
Contract		Event	Event	Event;
Contract	Issuing	Limit	Limit	Limit;
Contract		Order	Order	Order;
Contract		Preferred	Preferred	Preferred;
Contract		Relation	Relation	Relation;
Contract		Status	Status	Status;

Fig. 1. List of system objects

In the *Name* column, all Advanced Applications module objects are listed for which applications to create or change object parameters can be registered.

The *Object For Type* field specifies the system object to which the corresponding Advanced Applications module object belongs. For example, the ContractAddress object belongs to a contract registered in the system. The value in this field is used for filtering module objects when working with forms.

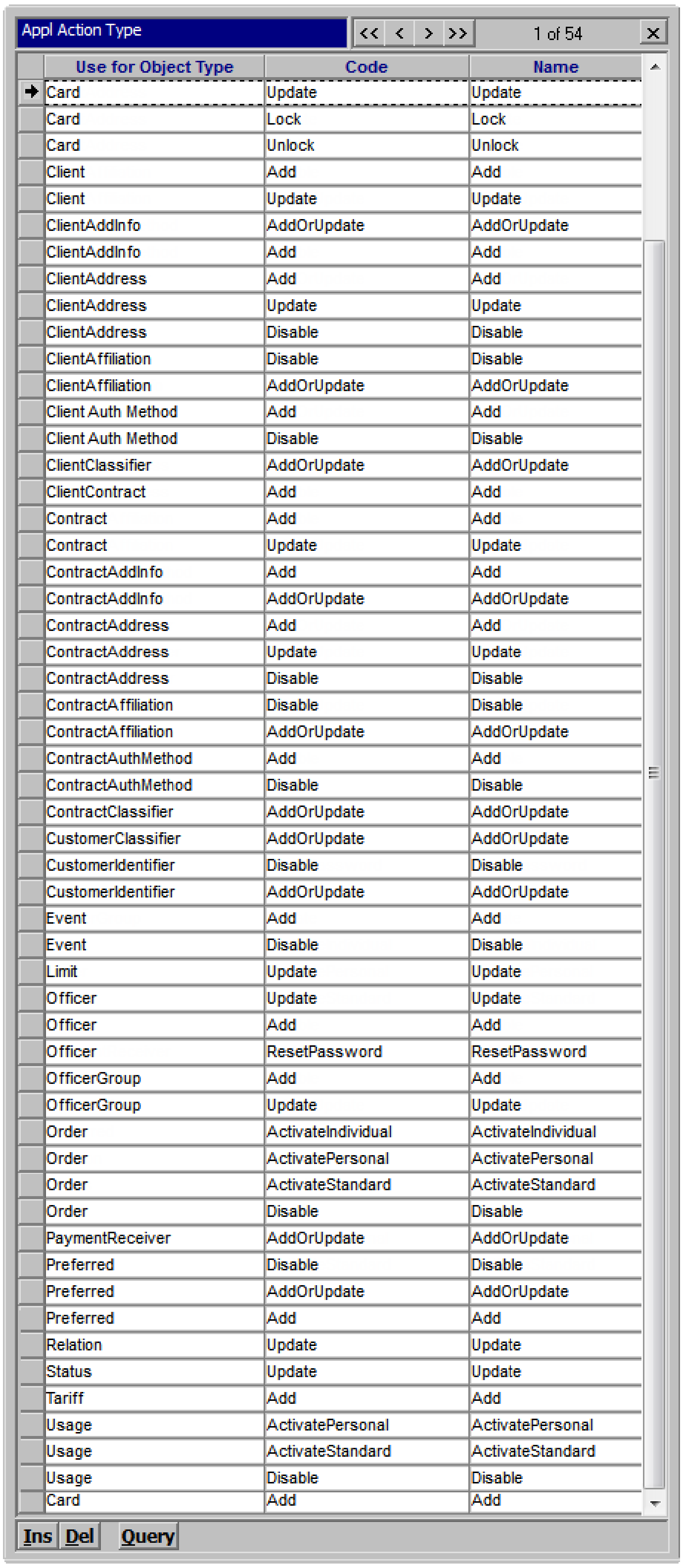
The *Product Category* field shows the category to which a given object belongs (Issuing or Acquiring).

The *Code* field contains the object identifier code (usually the value in this field corresponds to the value in the *Name* field).

The *Used Objects* field contains a list of objects actions on which can be provided for by an application. List elements are delimited by semi-colons.

Catalogue of Actions with Objects

In the catalogue of actions with objects (Advanced Applications R2 → Setup → Appl Action Type) all actions are listed that can be executed with system objects using the Advanced Applications module (see Fig. 2).



Use for Object Type	Code	Name
Card	Update	Update
Card	Lock	Lock
Card	Unlock	Unlock
Client	Add	Add
Client	Update	Update
ClientAddInfo	AddOrUpdate	AddOrUpdate
ClientAddInfo	Add	Add
ClientAddress	Add	Add
ClientAddress	Update	Update
ClientAddress	Disable	Disable
ClientAffiliation	Disable	Disable
ClientAffiliation	AddOrUpdate	AddOrUpdate
Client Auth Method	Add	Add
Client Auth Method	Disable	Disable
ClientClassifier	AddOrUpdate	AddOrUpdate
ClientContract	Add	Add
Contract	Add	Add
Contract	Update	Update
ContractAddInfo	Add	Add
ContractAddInfo	AddOrUpdate	AddOrUpdate
ContractAddress	Add	Add
ContractAddress	Update	Update
ContractAddress	Disable	Disable
ContractAffiliation	Disable	Disable
ContractAffiliation	AddOrUpdate	AddOrUpdate
ContractAuthMethod	Add	Add
ContractAuthMethod	Disable	Disable
ContractClassifier	AddOrUpdate	AddOrUpdate
CustomerClassifier	AddOrUpdate	AddOrUpdate
CustomerIdentifier	Disable	Disable
CustomerIdentifier	AddOrUpdate	AddOrUpdate
Event	Add	Add
Event	Disable	Disable
Limit	Update	Update
Officer	Update	Update
Officer	Add	Add
Officer	ResetPassword	ResetPassword
OfficerGroup	Add	Add
OfficerGroup	Update	Update
Order	ActivateIndividual	ActivateIndividual
Order	ActivatePersonal	ActivatePersonal
Order	ActivateStandard	ActivateStandard
Order	Disable	Disable
PaymentReceiver	AddOrUpdate	AddOrUpdate
Preferred	Disable	Disable
Preferred	AddOrUpdate	AddOrUpdate
Preferred	Add	Add
Relation	Update	Update
Status	Update	Update
Tariff	Add	Add
Usage	ActivatePersonal	ActivatePersonal
Usage	ActivateStandard	ActivateStandard
Usage	Disable	Disable
Card	Add	Add

Fig. 2. Actions with system objects

In the *Use for Object Type* column, all objects from the catalogue of Advanced Applications objects are listed (see Fig. 1).

Each row of the list corresponds to one action than can be executed with the corresponding object. The action is shown in the *Name* field.

⚠ Advanced Application module service catalogues may only be edited by WAY4 vendor representatives.

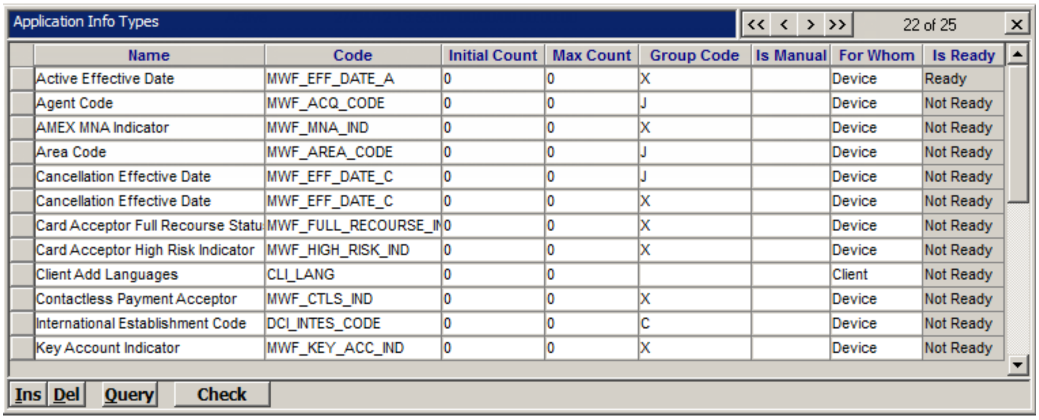
Application Additional Information Classifiers (Application Info Types)

For each application, additional information can be specified in the APPL_INFO special table. This information can be used to execute custom check or approve procedures.

Additional information is stored in the APPL_INFO table with the data type specified. Additional information data types are additional information classifiers (Application Info Type).

Data is entered in the APPL_INFO table using custom forms. These forms are configured using data filtering according to the value of the Application Info Type code (see the section "Form Builder Window: the "Fields" Tab" in the document "Form Builder")

Properties of application additional information types are specified in the "Application Info Type" form (see Fig. 3) (Advanced Applications R2 → Setup → Application Info Types).



Name	Code	Initial Count	Max Count	Group Code	Is Manual	For Whom	Is Ready
Active Effective Date	MWF_EFF_DATE_A	0	0	X		Device	Ready
Agent Code	MWF_ACQ_CODE	0	0	J		Device	Not Ready
AMEX MNA Indicator	MWF_MNA_IND	0	0	X		Device	Not Ready
Area Code	MWF_AREA_CODE	0	0	J		Device	Not Ready
Cancellation Effective Date	MWF_EFF_DATE_C	0	0	J		Device	Not Ready
Cancellation Effective Date	MWF_EFF_DATE_C	0	0	X		Device	Not Ready
Card Acceptor Full Recourse Statu	MWF_FULL_RECOURSE_IN	0	0	X		Device	Not Ready
Card Acceptor High Risk Indicator	MWF_HIGH_RISK_IND	0	0	X		Device	Not Ready
Client Add Languages	CLI_LANG	0	0			Client	Not Ready
Contactless Payment Acceptor	MWF_CTLS_IND	0	0	X		Device	Not Ready
International Establishment Code	DCI_INTES_CODE	0	0	C		Device	Not Ready
Key Account Indicator	MWF_KEY_ACC_IND	0	0	X		Device	Not Ready

Fig. 3. Form for specifying additional info type properties

This form contains the following fields:

- *Name* – classifier name.
- *Code* – classifier code; this value is used when creating an additional information input form. The combination of the *Code* and *Group Code* field values must be unique.
- *Initial Count* – the number of records in the application additional information table (APPL_INFO) that will be created when the form is opened for the first time.
- *Max Count* – the maximum number of records in the additional information table for one application of the given classifier type; this value can be controlled with custom procedures.
- *Group Code* – the classifier group code; this value is used to group application additional information.

- *Is Manual* – flag to indicate (YES value) that the corresponding record in the APPL_INFO table is created by the user; this field can be used to sort records in the APPL_INFO table.
- *For Whom* – the field specifies the object or process for which additional information is intended:
    - Client (C) – client record.
    - Contract (A) – contract.
    - Merchant Management (M) – merchant management.
    - Device Parm to Doc (D) – device parameters (tags) that must be saved in a document's *Add Info* field.
    - Device Parm Ctrl (S) – record of device parameters that affect document processing.

The [Check] button is used to call a procedure that validates the selected additional information data type.

Records are created in the additional information table and the correspondence of the number of records to set limits is checked according to standard system procedures.

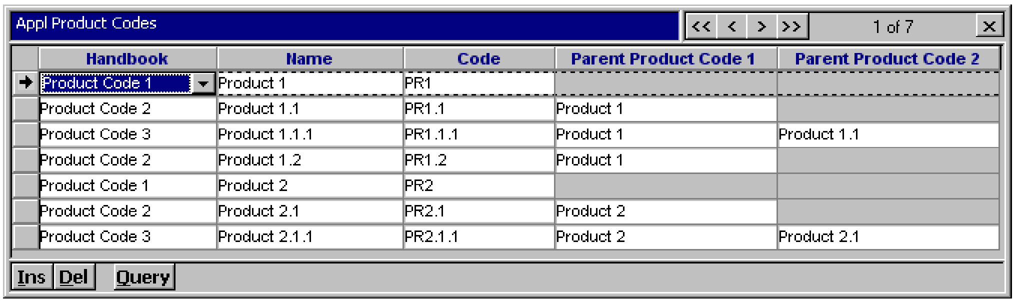
Product Codes

When entering an application to change or add a contract record in the Advanced Applications module, it is possible to specify the Product for the contract by selecting a Product code.

A Product is selected by specifying its code when there are a large number of Products registered in the system. When selecting a Product by code, only those Products for which a code is specified will be shown in the list, meaning Product codes are used for filtering the Product list.

To select a Product by code, codes must have been set earlier for Products used when entering applications (Advanced Applications R2 → Setup → Appl Product Code). The list of Product codes is organised as a hierarchy with three (or fewer) levels.

The list of Product codes is configured in the "Appl Product Codes" form (see Fig. 4).



Handbook	Name	Code	Parent Product Code 1	Parent Product Code 2
Product Code 1	Product 1	PR1		
Product Code 2	Product 1.1	PR1.1	Product 1	
Product Code 3	Product 1.1.1	PR1.1.1	Product 1	Product 1.1
Product Code 2	Product 1.2	PR1.2	Product 1	
Product Code 1	Product 2	PR2		
Product Code 2	Product 2.1	PR2.1	Product 2	
Product Code 3	Product 2.1.1	PR2.1.1	Product 2	Product 2.1

Fig. 4. Product codes

Fill in the following fields of this form:

- *Handbook* – drop-down list to select the Product code hierarchy level; this field can have one of the following values:
  - "Product Code 1" – the top level of the hierarchy. For a Product code with this value in the *Handbook* field:
    - Parent Product codes are not specified (values in the *Parent Product Code 1* and *Parent Product Code 2*) fields.
    - The value in the *Code* field must be the same as the value of the *Product Code* field set for the Product when configuring additional parameters (Full → Configuration Setup → Products → Product Definition → Products).
  - "Product Code 2" – the second level of the hierarchy. For a Product code with this value in the *Handbook* field:
    - The parent Product code (top level of the hierarchy) is specified in the *Parent Product Code 1* field.
    - The value in the *Code* field must be the same as the value of the *Code 2* field set for the Product when configuring additional parameters (Full → Configuration Setup → Products → Product Definition → Products).
  - "Product Code 3" – the third (lowest) level of the hierarchy. For a Product code with this value in the *Handbook* field:
    - The parent Product code (top level of the hierarchy) is specified in the *Parent Product Code 1* field.
    - The parent Product code (second level of the hierarchy) is specified in the *Parent Product Code 2* field.
    - The value in the *Code* field must be the same as the value of the *Code 3* field set for the Product when configuring additional parameters (Full → Configuration Setup → Products → Product Definition → Products).
- *Name* – Product code name.
- *Code* – Product code.
- *Parent Product Code 1* – field for selecting the parent Product code (top level of the hierarchy).
- *Parent Product Code 2* – field for selecting the Parent product code (second level of the hierarchy).

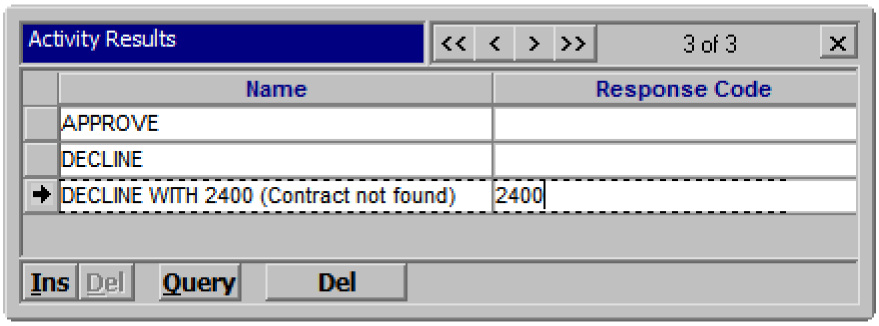
Chapter 3. Advanced Applications Module Configurations

When setting up the Advanced Applications module, it is possible to set a list of users and their privileges, and application workflow strategy.

Service Settings

In the current version of the module, the catalogue of workflow activity results is a service setting.

The catalogue of workflow activity results (Advanced Applications R2 → Setup → Workflow Configuration → OpenWay Settings → Activity Results) is configured for all domains (see the section "Domains").



	Name	Response Code
	APPROVE	
	DECLINE	
➔	DECLINE WITH 2400 (Contract not found)	2400

Fig. 5. Activity results

The name of the application processing result is specified in the *Name* field of the "Activity Results" form (see Fig. 5).

The error code for an application processing result can be specified in the *Response Code* field.

The application processing result can later be linked to a particular action (see the section Workflow Activities").

For actions whose execution results in a particular error code (*Response Code*), a special step can be set up when configuring rules in workflow strategies (see the section "Workflow Strategies").

Existing error codes are described in the section "Response File Error Messages" of the document "Importing and Exporting Advanced Applications (XML Format)."

By default, the "Activity Results" form (see Fig. 5) contains two standard records of application processing results. These records may not be edited:

- "Decline" – the action was executed with an error, the application was declined.
- "Approve" – the action was executed successfully, the application was approved.

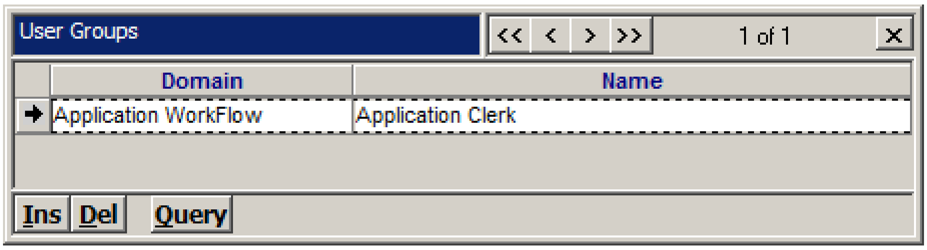
Activity results are used when creating rules required, in particular, to transfer applications from one step to another (see "Selecting a Strategy").

User Administration

Module users work within groups registered for a domain (see "Domains").

Users are distinguished by privileges to execute activities with applications. A set of user privileges (list of permitted activities) is determined by the role assigned to the user.

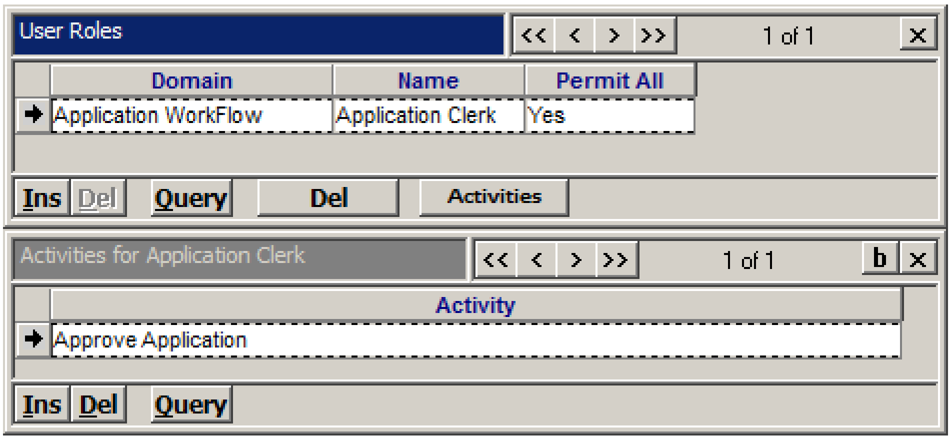
User groups are registered within domains using the "User Groups" form (see Fig. 6), opened by the "Advanced Applications R2 → Workflow Configuration → User Management → User Groups" menu item.



Domain	Name
Application WorkFlow	Application Clerk

Fig. 6. User groups

User roles are registered and configured using the "User Roles" form (see Fig. 7), opened by the "Advanced Applications R2 → Workflow Configuration → User Management → User Roles" menu item.



Domain	Name	Permit All
Application WorkFlow	Application Clerk	Yes

Activity
Approve Application

Fig. 7. Configuring user roles

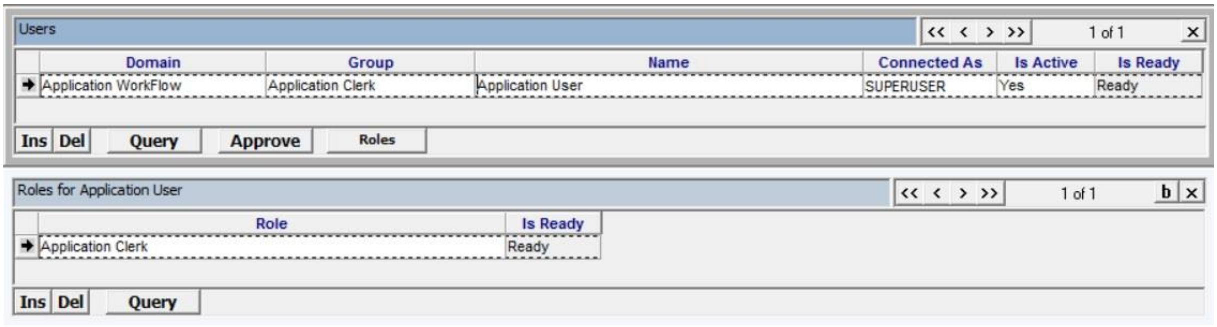
Roles are registered within a domain.

When the value of the *Permit All* field is set to "Yes", users with the given role may execute all activities with applications.

If the "No" value is set in this field, the role may be permitted to execute certain activities.

To assign permitted activities to a role, click the [Activities] button and in the "Activities for <name of role>" form, use the drop-down list to add the desired activities.

Users are registered in the "Users" form (see Fig. 8), opened by the "Advanced Applications R2 → Workflow Configuration → User Management → Users" menu item.



Domain	Group	Name	Connected As	Is Active	Is Ready
Application WorkFlow	Application Clerk	Application User	SUPERUSER	Yes	Ready

Role	Is Ready
Application Clerk	Ready

Fig. 8. User registration

To register a user, specify the domain and group in which the user will be included. In the *Connected As* field, specify the DB Manager user name under which the Advanced Applications module user will connect to the WAY4 database. To activate the user, specify "Yes" in the *Is Active* field (this field can also be used to deactivate the user).

When user parameters have been configured, click the [Approve] button to confirm the changes.

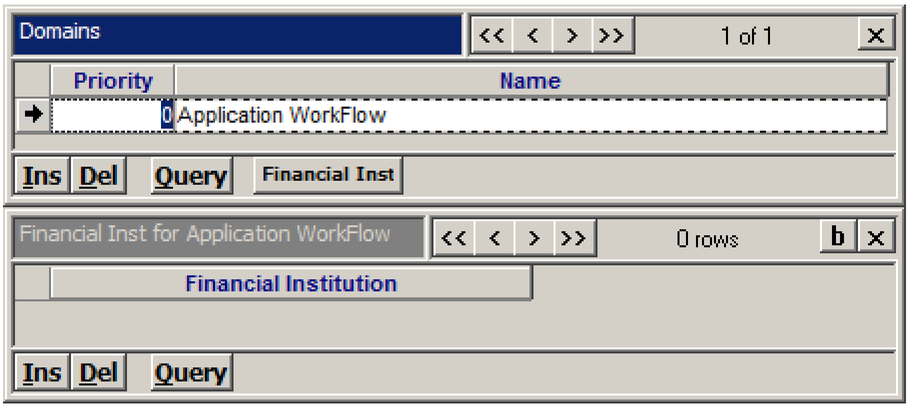
The user will be registered and activated (*Is Ready* = "Ready" and *Is Active* = "Yes").

Domains

The use of domains in the Advanced Applications module ensures application grouping by financial institutions.

Application workflow strategies are registered and used within a domain.

Domains are registered and configured in the "Domains" form (see Fig. 9), opened by the "Advanced Applications R2 → Workflow Configuration → Domains" menu item.



Priority	Name
0	Application WorkFlow

Financial Institution

Fig. 9. Form for registering and configuring domains

In the *Priority* field, specify the value determining the procedure for selecting domains (the lower the value, the earlier the domain will be checked).

Clicking the [Financial Inst] button opens the form to specify the financial institution to which the domain belongs.

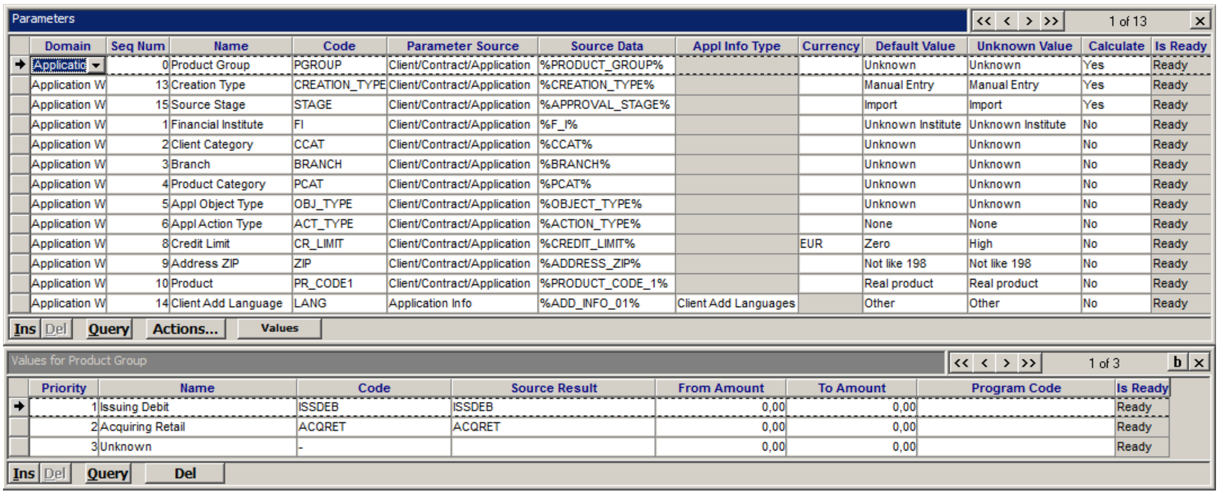
If a domain for the required financial institution is not found during selection, a domain will be chosen for which no financial institution is specified (if such a domain is configured).

Application Parameters

Application parameters are used by the module when selecting a workflow strategy and assigning a user (see "Selecting a Strategy").

Application parameters must have a value from the list of possible values. A value is assigned to a parameter after comparing the result of estimating (calculating) its value with the list of possible values.

Parameters are registered and configured in the "Parameters" form (see Fig. 10), opened by the "Advanced Applications R2 → Workflow Configuration → Parameters" menu item.



Domain	Seq Num	Name	Code	Parameter Source	Source Data	Appl Info Type	Currency	Default Value	Unknown Value	Calculate	Is Ready
Applicatio	0	Product Group	PGROUP	Client/Contract/Application	%PRODUCT_GROUP%			Unknown	Unknown	Yes	Ready
Application W	13	Creation Type	CREATION_TYPE	Client/Contract/Application	%CREATION_TYPE%			Manual Entry	Manual Entry	Yes	Ready
Application W	15	Source Stage	STAGE	Client/Contract/Application	%APPROVAL_STAGE%			Import	Import	Yes	Ready
Application W	1	Financial Institute	FI	Client/Contract/Application	%F_I%			Unknown Institute	Unknown Institute	No	Ready
Application W	2	Client Category	CCAT	Client/Contract/Application	%CCAT%			Unknown	Unknown	No	Ready
Application W	3	Branch	BRANCH	Client/Contract/Application	%BRANCH%			Unknown	Unknown	No	Ready
Application W	4	Product Category	PCAT	Client/Contract/Application	%PCAT%			Unknown	Unknown	No	Ready
Application W	5	Appl Object Type	OBJ_TYPE	Client/Contract/Application	%OBJECT_TYPE%			Unknown	Unknown	No	Ready
Application W	6	Appl Action Type	ACT_TYPE	Client/Contract/Application	%ACTION_TYPE%			None	None	No	Ready
Application W	8	Credit Limit	CR_LIMIT	Client/Contract/Application	%CREDIT_LIMIT%		EUR	Zero	High	No	Ready
Application W	9	Address ZIP	ZIP	Client/Contract/Application	%ADDRESS_ZIP%			Not like 198	Not like 198	No	Ready
Application W	10	Product	PR_CODE1	Client/Contract/Application	%PRODUCT_CODE_1%			Real product	Real product	No	Ready
Application W	14	Client Add Language	LANG	Application Info	%ADD_INFO_01%	Client Add Languages		Other	Other	No	Ready

Priority	Name	Code	Source Result	From Amount	To Amount	Program Code	Is Ready
1	Issuing Debit	ISSDEB	ISSDEB	0,00	0,00		Ready
2	Acquiring Retail	ACQRET	ACQRET	0,00	0,00		Ready
3	Unknown	-		0,00	0,00		Ready

Fig. 10. Configuring application parameters

The *Seq Num* field is used to sort values in this form.

The *Parameter Source* field specifies an object containing the parameter:

- Client/Contract/Application – the application
- Application Info – the record in the APPL_INFO table of additional information
- Application Add Data – the ADD_DATA field of the application.

The *Source Data* field specifies the type of data received from the object (application field).

- The following data types may be specified for application fields:
  - %APPLICATION_TYPE% – application type.
  - %SOURCE_CODE% – code of the system in which an application was created.
  - %DUPLICATE_COUNT% – number of duplicate applications.

- %CREATION_TYPE% – the way the application is created. Possible values are:
  - ♦ ENTRY – manual entry (screen forms).
  - ♦ IMPORT – import from a file using an import pipe.
  - ♦ WS – entry using web services (automatic creation of applications when working with customer service workbench, merchant service workbench and merchant portal).
  - ♦ SPLIT – the application is created as a subordinate application, later subject to deferred acceptance (see the section "Accepting Subordinate Applications" of the document "Advanced Applications R2").
-
- %OBJECT_TYPE% –application object type (see "Catalogue of Objects").
- %ACTION_TYPE% – action with an application object (see "Catalogue of Actions with Objects").
- %F_I% – client financial institution.
- %BRANCH% – client branch.
- %CCAT% – client category.
- %CLIENT_TYPE% – client type.
- %CLASSIFIER_TYPE% – type of client/contract classifier. When configuring parameter values, note that if an application contains several sections for working with classifiers, all classifier type codes will be grouped in a list (with commas to separate them) and compared with the value specified in the *Source Result* field (see the section "Case Parameters" of the document "WAY4 Case Management Configuration"). To determine the presence of sections with the specified classifier type in an application, it is recommended to specify the following values in the *Program Code* field:
  - ♦ *Program Code* = "INCLUDE" – the parameter value will be returned if the value specified in the *Source Result* field is included in the list of classifier types.
  - ♦ *Program Code* = "EXCLUDE" – the parameter value will be returned if the value specified in the *Source Result* field is not included in the list of classifier types.
- %PCAT% – contract Product category.
- %LANGUAGE% – client language set in the application.
- %AGE% – client age calculated according to the date of birth specified in the application.
- %CREDIT_LIMIT% – credit limit.
- %ADDRESS_ZIP% – postal (ZIP) code (part of address) specified in the application.

  - %PRODUCT_CODE_1%, %PRODUCT_CODE_2%, %PRODUCT_CODE_3% – Product code.
  - %PRODUCT_GROUP% – code of the group to which the Product for the contract specified in the application belongs.
- For records in the APPL_INFO table of additional information: %ADD_INFO_01% – the value from the specified field of the APPL_INFO table of additional information with the data type specified in the *Appl Info Type* field.
- For the ADD_DATA field of the application: %TAG NAME%. Moreover, in the "Values for <parameter name>" form, a record must be created for each of the possible tag values. This data type may be used to select a workflow strategy.

The *Calculate* field specifies whether this parameter is calculated for the application.

The *Default Value* field specifies the parameter value used by default when it was not possible to obtain an estimation (calculation) result.

The *Unknown Value* field specifies the parameter value if it was not possible to associate the result of determining the value with any of the possible values.

Possible parameter values are set in the "Values for <name of parameter>" form, opened by clicking the [Values] button.

This form is used for parameters with text or numeric values.

The *Source Result* field specifies text with which the parameter value is compared.

The *From Amount* and *To Amount* fields specify the lower and upper value specify the lower and upper values of the range of numeric parameter values.

The following procedure is used to determine parameter values.

The estimation of a text parameter value is compared with the value in the *Source Result* field. If correspondence is determined, the parameter is assigned the value contained in the *Name* field of the string found.

The estimation of a numeric parameter value is compared with the value in the *From Amount* and *To Amount* fields. If a value is found to fall in this range, the parameter is assigned the value contained in the *Name* field of the string found.

A correspondence search is made in the order specified by the value of the *Priority* field (the lower the value, the earlier correspondence is checked).

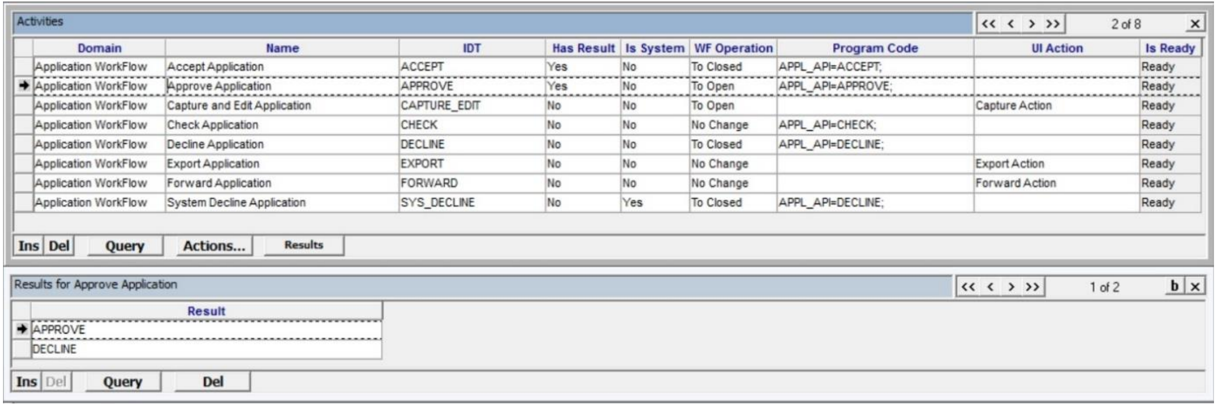
If no correspondence with the estimated (calculated) value was found in the list of possible parameter values, the parameter is assigned the value specified in the *Unknown Value* field.

After configuration is completed, confirm the changes by clicking the [Actions…] button and selecting the "Approve" or "Approve All" item from the context menu. Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log.

To delete a parameter from the list, in the strategy schemes form click the [Actions…] button and select "Del" from the context menu.

Workflow Activities

The catalogue of activities that can be executed by a user or the system during application workflow is configured in the "Activities" form (see Fig. 11), opened by the "Advanced Applications R2 → Workflow Configuration → Activities" menu item.



Activities

Domain	Name	IDT	Has Result	Is System	WF Operation	Program Code	UI Action	Is Ready
Application WorkFlow	Accept Application	ACCEPT	Yes	No	To Closed	APPL_API=ACCEPT;		Ready
Application WorkFlow	Approve Application	APPROVE	Yes	No	To Open	APPL_API=APPROVE;		Ready
Application WorkFlow	Capture and Edit Application	CAPTURE_EDIT	No	No	To Open		Capture Action	Ready
Application WorkFlow	Check Application	CHECK	No	No	No Change	APPL_API=CHECK;		Ready
Application WorkFlow	Decline Application	DECLINE	No	No	To Closed	APPL_API=DECLINE;		Ready
Application WorkFlow	Export Application	EXPORT	No	No	No Change		Export Action	Ready
Application WorkFlow	Forward Application	FORWARD	No	No	No Change		Forward Action	Ready
Application WorkFlow	System Decline Application	SYS_DECLINE	No	Yes	To Closed	APPL_API=DECLINE;		Ready

Results for Approve Application

Result
APPROVE
DECLINE

Fig. 11. Configuring workflow activities

The parameters of activities that can be executed when an application is processed are specified in this form.

The *Has Result* field specifies whether the respective activity is characterized by a result. If the "Yes" value is set in this field, the [Results] button will become available for the activity. This button opens the form for specifying possible results of the activity's execution. Results of activity execution are selected from the "Activity Results" catalogue (see the section "Service Settings").

The *Is System* field specifies if an activity is executed by the system or by a user. For system activities (the "Yes" value), before executing an activity, user privileges for executing this activity are not checked.

The *WF Operation* field defines the status of the application workflow process after an activity has been performed:

- "To Closed" – close the application workflow process.
- "To Open" – reopen the application workflow process.
- "No Change" – do not change the status of the application workflow process.

In a standard strategy, "To Closed" is set for activities to accept or decline applications ("Accept Application", "Decline Application" and "System Decline Application"). "To Open" is set for activities to approve or edit an application ("Approve Application" and "Capture and Edit Application").

After changing application workflow settings, it is recommended to close all processed applications as a group (applications with the "Posted" and "Declined" values). To do so, use the menu item "OpenWay → Advanced Applications R2 → Application Processing → Close Posted Applications".

To improve performance, it is recommended to run this process with the "PARALLEL" parameter (see the document "Running WAY4™ Processes in Parallel").

The Program Code field contains the code for calling an API procedure. The value in this field may only be modified by WAY4 vendor representatives.

Depending on the value specified in the Program Code field, non-system activities (with the "No" value in the Is System field) may be available to the user when working with various elements of the Advanced Applications module user interface.

The UI Action field makes it possible to specify the button (or other interface element) that will call execution of this action. The field may contain the following values:

- "Capture Action" – makes it possible to call the specified action from Advanced Applications R2 module screen forms by clicking the [Actions…] button, and subsequently selecting the "Capture" menu item (or by clicking the [Capture] button).
- "Export Action" – makes it possible to call the specified action by selecting the user menu item "Advanced Applications R2 → Application Processing → XML Applications Export".
- "Forward Action" – makes it possible to call an action from Advanced Applications R2 screen forms by selecting the "Forward" context menu item called by clicking the [Actions…} button. When this item is selected, a window opens for forwarding the application. This window contains the list of all actions available for this strategy step for which the "Forward Action" value was specified in the *UI Action* field during configuration.

Usually, for an action record, it is recommended to specify a value either in the *Program Code* field or in the *UI Action* field.

Table 1 shows the correspondence between values in the *Program Code* or *UI Action* fields of action records and interface elements in which these actions may be available.

Table 1. Availability of actions in the Advanced Applications R2 module user interface depending on values in the Program Code or UI Action field

Value in the ***Program Code*** or ***UI Action*** field	Name of the workflow activity	Interface element in which the activity is available	User menu item or button
Program Code = APPL_API= ACCEPT	Accept an application (see the section "Accepting Applications" of the document "Advanced Applications R2)"	User menu	Accept processed issuing module applications: "Advanced Applications R2 → Application Processing → Issuing Applications – Accept"; Accept processed acquiring module applications: "Advanced Applications R2 → Application Processing → Acquiring Application – Accept".
Program Code = APPL_API= APPROVE	Approve an application (see the section "Approving Applications" of the document "Advanced Applications R2")	• Application input form. • Form for manually processing applications.	The "Approve" item of the context menu opened by clicking the [Actions…] button.
Program Code = APPL_API= CHECK	Check an application (see the section "Checking Applications" of the document "Advanced Applications R2")	• Application input form. • Form for manually processing applications.	The "Check" item of the context menu opened by clicking the [Actions…] button.
Program Code = APPL_API= DECLINE	Decline an application (see the section "Declining Applications" of the document "Advanced Applications R2")	Form for manually processing applications.	The "Decline" item of the context menu opened by clicking the [Actions…] button.
UI Action = "Forward Action"	Forward an application (see the section "Forwarding Applications" of the document Advanced Applications R2")	Form for manually processing applications.	The "Forward" item of the context menu opened by clicking the [Actions…] button. When this item is selected, a window for forwarding the application opens. This window contains a list of all activities with the "FORWARD" value in the *Program Code* field that are available for this strategy step.
UI Action = "Capture Action"	"Captures" an application and opens the application form in editing mode (see the sections "Manually Processing Applications" and "Processing Declined Applications" of the document "Advanced Applications R2")	The list of imported applications, approved applications and the list of declined applications.	The [Actions…] button, with subsequent selection of the "Capture" menu item, or the [Capture] button.

Value in the *Program Code* or *UI Action* field	**Name of the workflow activity**	**Interface element in which the activity is available**	**User menu item or button**
UI Action = "Export Action"	Application Export (see the section "Exporting Applications" of the document "Advanced Applications R2")	User menu	"Advanced Applications R2 → Application Processing → XML Applications Export".

The activities listed above are available in the specified screen forms if they have been added to the list of activities permitted for the strategy according to which the application is processed, or to the list of activities available for a specific strategy step (see the section "Workflow Strategies").

After configuration is completed, confirm the changes by clicking the [Actions…] button and selecting "Approve" from the context menu. Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log.

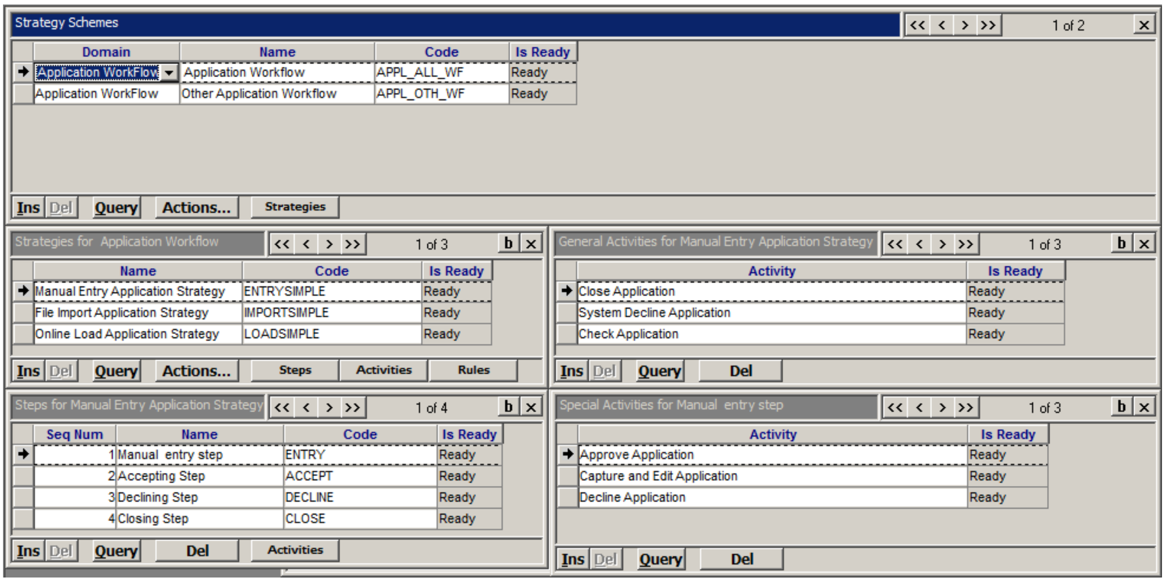
To delete activities from the list, click the [Actions…] button and select "Delete" from the context menu.

Workflow Strategies

Workflow strategies are registered within domains (see the section "Domains") and are grouped in schemes related to, for example, issuing, acquiring, etc.

To configure workflow strategies, select the "Advanced Applications R2 → Workflow Configuration → Strategy Schemes" menu item.

This will open the "Strategy Schemes" form together with a group of child forms (see Fig. 12).



Strategy Schemes

Domain	Name	Code	Is Ready
Application WorkFlow	Application Workflow	APPL_ALL_WF	Ready
Application WorkFlow	Other Application Workflow	APPL_OTH_WF	Ready

Strategies for Application Workflow

Name	Code	Is Ready
Manual Entry Application Strategy	ENTRYSIMPLE	Ready
File Import Application Strategy	IMPORTSIMPLE	Ready
Online Load Application Strategy	LOADSIMPLE	Ready

General Activities for Manual Entry Application Strategy

Activity	Is Ready
Close Application	Ready
System Decline Application	Ready
Check Application	Ready

Steps for Manual Entry Application Strategy

Seq Num	Name	Code	Is Ready
1	Manual entry step	ENTRY	Ready
2	Accepting Step	ACCEPT	Ready
3	Declining Step	DECLINE	Ready
4	Closing Step	CLOSE	Ready

Special Activities for Manual entry step

Activity	Is Ready
Approve Application	Ready
Capture and Edit Application	Ready
Decline Application	Ready

1 of 2
Ins Del Query Actions... Strategies
1 of 3
Ins Del Query Actions... Steps Activities Rules
1 of 3
Ins Del Query Del
1 of 4
Ins Del Query Del Activities
1 of 3
Ins Del Query Del

Fig. 12. Form for configuring workflow strategies

In the "Strategy Schemes" form, schemes used to group workflow strategies are registered within domains.

Strategies included in a scheme are registered in the child form "Strategies for <name of scheme>", opened by clicking the [Strategies] button in the parent form.

The following are configured for strategies:

- Activities that can be executed during workflow within the strategy (executed in all steps of the strategy).
- Workflow steps.

To configure activities, click the [Activities] button in the strategies form to open the child form "General Activities for <name of strategy>". Activities that can be added to a strategy are selected from a list created earlier (see "Workflow Activities").

To create a copy of the strategy, in the strategies form click the [Actions…] button and select "Duplicate" from the context menu.

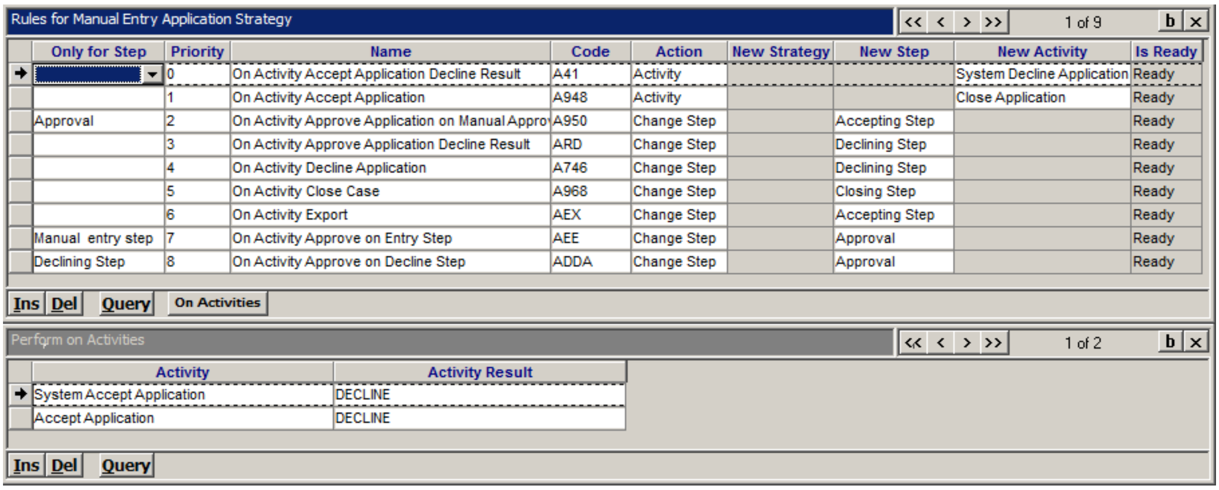
To delete a strategy from the list, in the strategies form, click the [Actions…] button and select "Delete" from the context menu.

To configure workflow steps, click the [Steps] button in the strategies form to open the child form "Steps for <name of strategy>". In this form, the user creates the sequence of workflow steps, specifying the sequence number of the step in the *Seq Num* field.

For each registered step, an individual set of activities that can be executed when the application is located at this step may be configured. This setting is made in the same way as for strategies, in the "Special Activities for <name of step>" form opened by clicking the "Activities" button.

An application is transferred from one step to another within a workflow strategy, or transferred to another strategy when rules configured for the strategy or step are fulfilled.

To configure rules, for the selected strategy in the form "Strategies for <name of scheme>", click the [Rules] button to open the child form "Rules for <name of strategy>" (see Fig. 13).



Rules for Manual Entry Application Strategy

Only for Step	Priority	Name	Code	Action	New Strategy	New Step	New Activity	Is Ready
	0	On Activity Accept Application Decline Result	A41	Activity			System Decline Application	Ready
	1	On Activity Accept Application	A948	Activity			Close Application	Ready
Approval	2	On Activity Approve Application on Manual Appro	A950	Change Step		Accepting Step		Ready
	3	On Activity Approve Application Decline Result	ARD	Change Step		Declining Step		Ready
	4	On Activity Decline Application	A746	Change Step		Declining Step		Ready
	5	On Activity Close Case	A968	Change Step		Closing Step		Ready
	6	On Activity Export	AEX	Change Step		Accepting Step		Ready
Manual entry step	7	On Activity Approve on Entry Step	AEE	Change Step		Approval		Ready
Declining Step	8	On Activity Approve on Decline Step	ADDA	Change Step		Approval		Ready

Perform on Activities

Activity	Activity Result
System Accept Application	DECLINE
Accept Application	DECLINE

Fig. 13. Configuring rules

In the Advanced Applications module a rule is a condition and the result of the condition being met.

A rule's condition is the execution of a workflow activity, including with a specified result. Conditions for configurable rules are set in the child form "Perform on Activities", opened by clicking the [On Activities] button.

In this form's *Activity* field, specify the activities whose execution is a condition for the rule to be applied. In the *Activity Result* field, the result of executing the activity may be specified. If not, execution of the activity with any result will be considered a condition for applying the rule.

When configuring conditions, workflow activities are selected from an earlier configured list. Results of executing activities are selected from a list corresponding to the parameters of the configured activity (see "Workflow Activities").

The result of meeting a rule's conditions is specified in the *Action* field of the form for configuring rules. Possible values are:

- "Next Step" – go to the next step of the strategy.
- "Change Step" – go to the strategy step specified in the *New Step* field.
- "Change Strategy" – go to the first step of the strategy specified in the *New Strategy* field.
- "Activity" – execute the activity specified in the *New Activity* field.

The *Priority* field specifies the value determining the order of checking rules (the lower the value, the earlier the rule is checked).

A strategy step can be specified in the *Only for Step* field. If this field is filled in, the rule will be in effect only at the selected step. If the field is left empty, the rule will be applicable for the entire strategy, meaning conditions of the rule will be checked at each step.

When configuration is completed, to confirm the changes, click the [Actions…] button in the strategy schemes form and select "Approve" from the context menu. Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log.

To create a copy of a configured strategy scheme, click the [Actions…] button in the strategy scheme and select "Duplicate" from the context menu.

To delete a scheme from the list, click the [Actions…] button in the strategy schemes form and select "Delete" from the context menu.

Selecting a Strategy

When an application is registered, the system selects a workflow strategy.

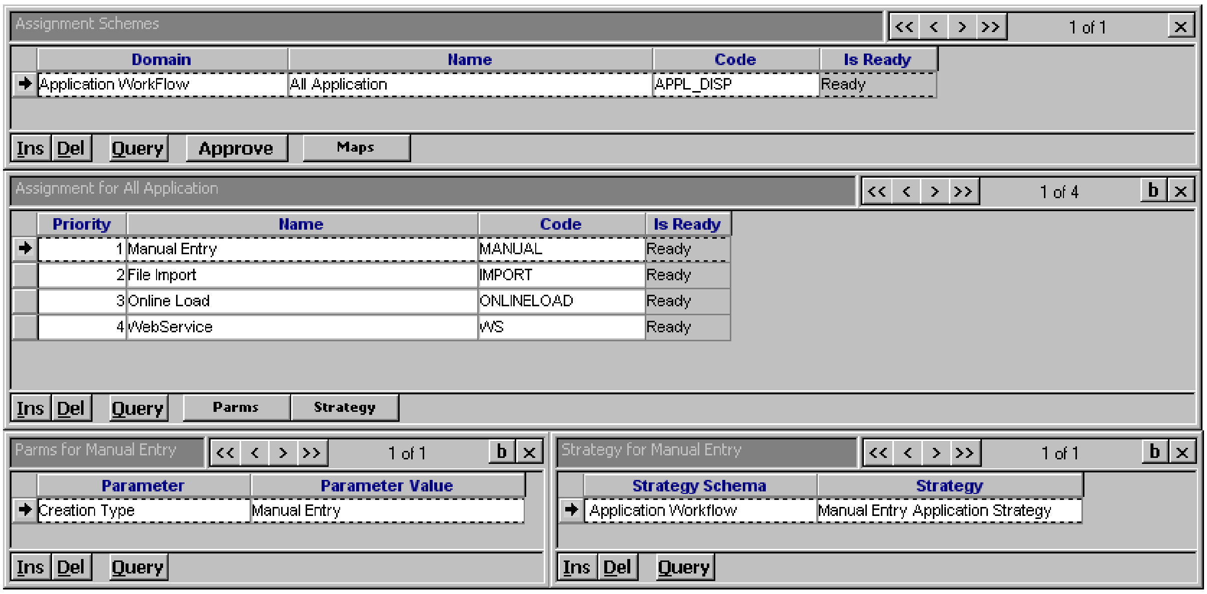
These actions are executed on the basis of application parameters. In the Advanced Applications module, sets of parameters are configured with specified values; these sets are called parameter maps.

During configuration, a specific workflow strategy is specified for each parameter map.

If the values of an application's parameters correspond to the values set during map configuration, the strategy specified for the map is selected for the application workflow.

To configure parameter maps, select the menu item "Advanced Applications R2 → Workflow Configuration → Assignment Schemes".

The "Assignment Schemes" form together with a group of child forms will open on the screen (see Fig. 14).



Assignment Schemes — 1 of 1

Domain	Name	Code	Is Ready
Application WorkFlow	All Application	APPL_DISP	Ready

Assignment for All Application — 1 of 4

Priority	Name	Code	Is Ready
1	Manual Entry	MANUAL	Ready
2	File Import	IMPORT	Ready
3	Online Load	ONLINELOAD	Ready
4	WebService	WS	Ready

Parms for Manual Entry — 1 of 1

Parameter	Parameter Value
Creation Type	Manual Entry

Strategy for Manual Entry — 1 of 1

Strategy Schema	Strategy
Application Workflow	Manual Entry Application Strategy

Fig. 14. Configuring parameter maps

Parameter maps are grouped in schemes registered in the "Assignment Schemes" form.

Parameter maps are configured in the "Assignment for <name of scheme>" form opened from the "Assignment Schemes" form by clicking the [Maps] button.

A set of parameters with specified values corresponds to each map, as well as a workflow strategy.

ⓘ If different parameters are specified in one card, they are joined through the logical "AND" operator, different values of the same parameter are joined through the logical "OR" operator.

When selecting a parameter map for an application, a check is made in the ascending order of priorities specified in the *Priority* field.

Parameters of maps are configured and strategies are specified in the child forms opened by clicking the [Parms] and [Strategy] buttons, respectively.

After configuring parameter maps, click the [Approve] button in the "Assignment Schemes" parent form to confirm the changes. Additional information about the results of approval can be obtained after selection in the user menu item "Full → Process Log → Process Log.

Chapter 4. Configuring Application Schemes

An application scheme makes it possible to set a main object (client or contract) for which an application is intended, and a hierarchy of objects related to it.

A scheme consists of sections, contains a main section (to enter data about the main object) and a set of subsections organized in a hierarchy. Each section is used to enter information about a system object (client, contract, address, etc.). A scheme may consist of one section.

Further when a user creates an application of a selected type according to a scheme, an object structure will be automatically created for the user to enter information about each object.

ℹ For a number of objects, information can be added even when the application scheme does not have a section for working with this object (see "Rules for Building Schemes").

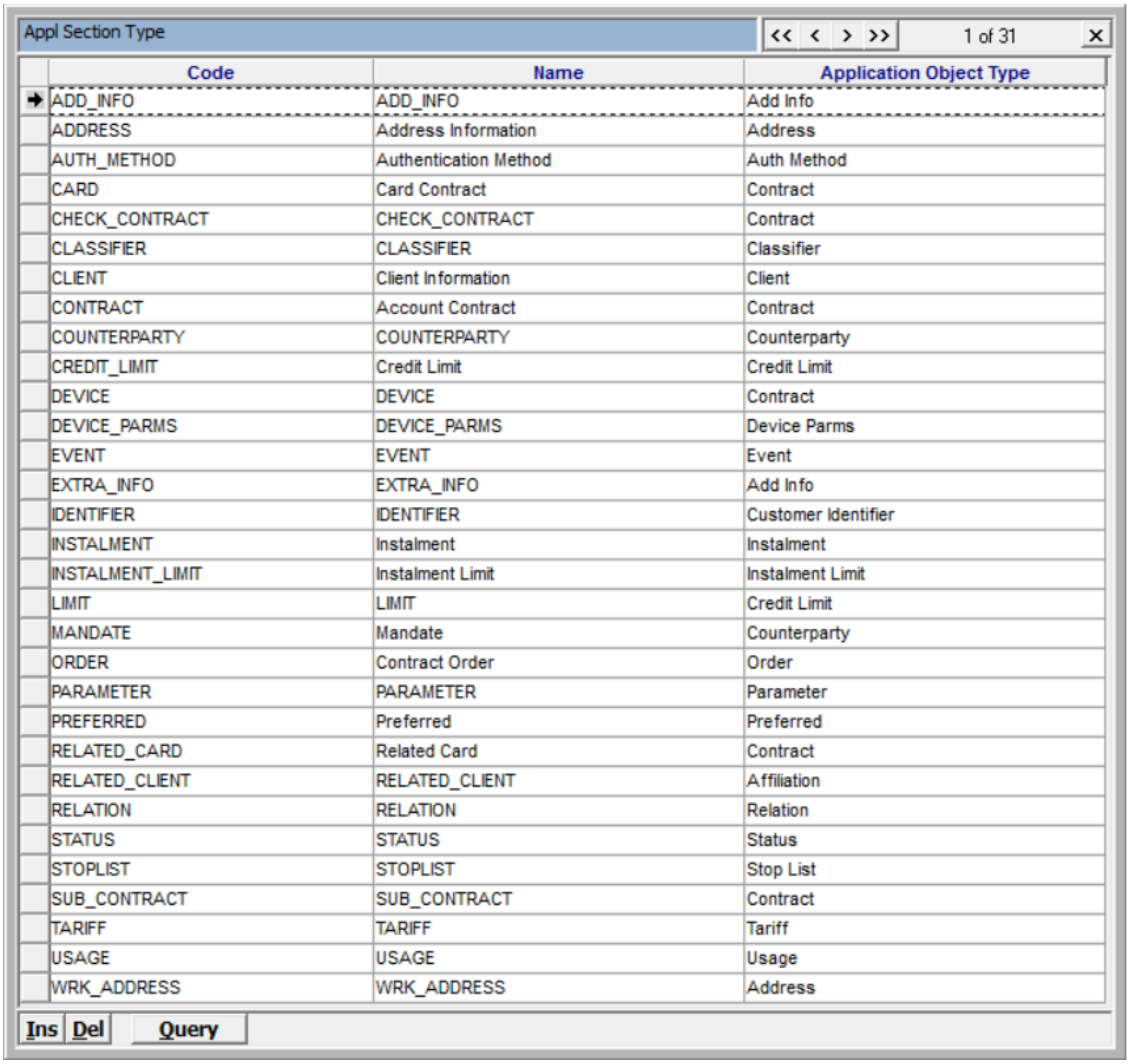
Example of a hierarchical object structure:

- Client record (main object):
  - Contract.
    - Address.
    - Card contract.

Application Section Types

An application scheme is built from sections of various type (see "Application Schemes").

Section types are registered and configured in the "Appl Section Type" form (see Fig. 15) opened by the menu item "Advanced Applications R2 → Setup → Appl Section Type".



Code	Name	Application Object Type
ADD_INFO	ADD_INFO	Add Info
ADDRESS	Address Information	Address
AUTH_METHOD	Authentication Method	Auth Method
CARD	Card Contract	Contract
CHECK_CONTRACT	CHECK_CONTRACT	Contract
CLASSIFIER	CLASSIFIER	Classifier
CLIENT	Client Information	Client
CONTRACT	Account Contract	Contract
COUNTERPARTY	COUNTERPARTY	Counterparty
CREDIT_LIMIT	Credit Limit	Credit Limit
DEVICE	DEVICE	Contract
DEVICE_PARMS	DEVICE_PARMS	Device Parms
EVENT	EVENT	Event
EXTRA_INFO	EXTRA_INFO	Add Info
IDENTIFIER	IDENTIFIER	Customer Identifier
INSTALMENT	Instalment	Instalment
INSTALMENT_LIMIT	Instalment Limit	Instalment Limit
LIMIT	LIMIT	Credit Limit
MANDATE	Mandate	Counterparty
ORDER	Contract Order	Order
PARAMETER	PARAMETER	Parameter
PREFERRED	Preferred	Preferred
RELATED_CARD	Related Card	Contract
RELATED_CLIENT	RELATED_CLIENT	Affiliation
RELATION	RELATION	Relation
STATUS	STATUS	Status
STOPLIST	STOPLIST	Stop List
SUB_CONTRACT	SUB_CONTRACT	Contract
TARIFF	TARIFF	Tariff
USAGE	USAGE	Usage
WRK_ADDRESS	WRK_ADDRESS	Address

Fig. 15. Section types

The *Code* field contains the section type code.

The *Name* field specifies the application section type name shown when entering and processing an application.

The *Application Object Type* field specifies the type of object for which this section type is intended.

ⓘ Special buttons in application processing forms are used to work with a number of application objects. The buttons will only be shown if section types with the corresponding value in the *Code* field are configured. These sections types and their codes are shown in Fig. 15. It is not recommended to change or delete these values.

Application Schemes

The "Appl Type" form (see Fig. 16) is used to work with application types. This form is opened by the "Advanced Applications R2 → Setup → Appl Type" menu item.

An application scheme is configured for each application type.

A scheme consists of sections. It contains a main section (for entering data about the main object) and a set of subsections, organized in a hierarchy. The main section of the scheme can be used to work with the "Client" or "Contract" object.

Each section is used to enter information about a system object (client, contract, address, etc.).

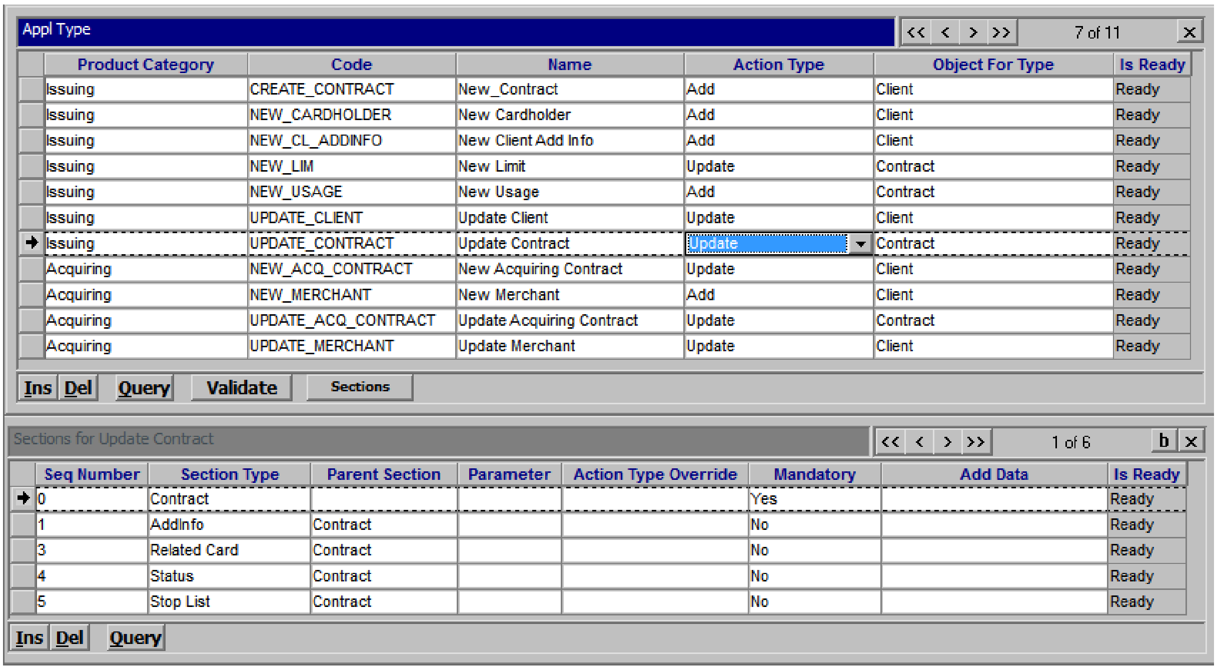
Each row in the "Appl Type" form (see Fig. 16) corresponds to an application type. When a type is created, the main object type of the application is specified for it.

All other properties of a type are specified when configuring the applications scheme.

Example of a hierarchical structure of objects:

- Client record (main object).
    - ▪ Contract.
        - ♦ Address.
        - ♦ Card contract.

For a number of objects, it is possible to add information event when the section intended for working with this object are missing from the application scheme (see "Rules for Building Schemes").



	Product Category	Code	Name	Action Type	Object For Type	Is Ready
	Issuing	CREATE_CONTRACT	New_Contract	Add	Client	Ready
	Issuing	NEW_CARDHOLDER	New Cardholder	Add	Client	Ready
	Issuing	NEW_CL_ADDINFO	New Client Add Info	Add	Client	Ready
	Issuing	NEW_LIM	New Limit	Update	Contract	Ready
	Issuing	NEW_USAGE	New Usage	Add	Contract	Ready
	Issuing	UPDATE_CLIENT	Update Client	Update	Client	Ready
➜	Issuing	UPDATE_CONTRACT	Update Contract	Update	Contract	Ready
	Acquiring	NEW_ACQ_CONTRACT	New Acquiring Contract	Update	Client	Ready
	Acquiring	NEW_MERCHANT	New Merchant	Add	Client	Ready
	Acquiring	UPDATE_ACQ_CONTRACT	Update Acquiring Contract	Update	Contract	Ready
	Acquiring	UPDATE_MERCHANT	Update Merchant	Update	Client	Ready

	Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
➜	0	Contract				Yes		Ready
	1	AddInfo	Contract			No		Ready
	3	Related Card	Contract			No		Ready
	4	Status	Contract			No		Ready
	5	Stop List	Contract			No		Ready

Fig. 16. Application schemes

The *Product Category* field contains the category to which the application type belongs (Issuing or Acquiring).

The *Name* field specifies the application type name.

The *Code* field specifies the application type code.

The *Action Type* field contains the type of action with the application object.

The *Object For Type* field contains the type of main object for which the application is used: "Client" or "Contract". The field may be empty if an application scheme is set up to create a hierarchy of contracts with different clients (see "Examples of Creating Application Schemes with a Hierarchy of Contracts for Different Clients").

The *Is Ready* field indicates whether the application type is ready for use. The value of the field will be "Yes" after the application type has been approved.

The {Validate} command is used to validate the scheme belonging to the selected application type. If errors are found during validation, the corresponding message will be displayed.

ℹ It is not recommended to delete the "New Cardholder" and "New Merchant" types from the list, or to change their codes.

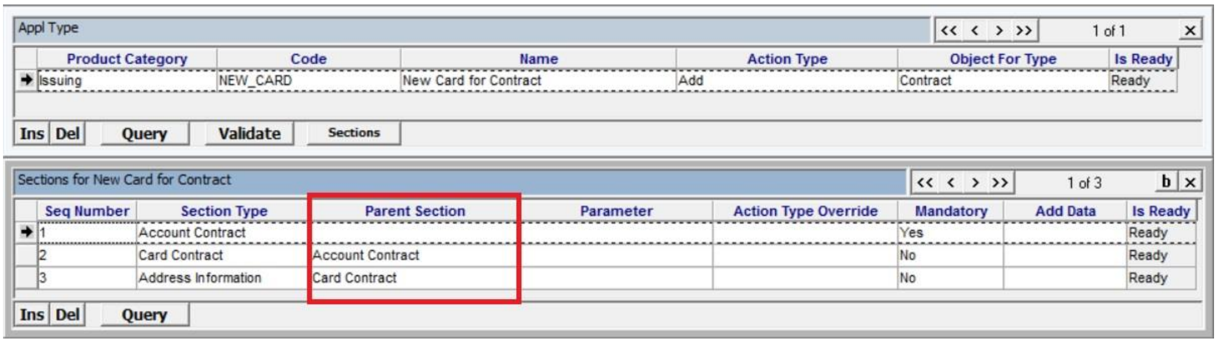
Work with an application scheme takes place in the "Sections for <scheme name>" child form opened by clicking the [Sections] button in the parent form (see Fig. 16).

The form contains the following fields:

- *Seq Number* – sequence number of a section in the scheme; the field must contain a value that is unique in the scheme.
- *Section Type* – section type (see "Application Section Types"). The main section of the scheme must be used to work with either a "Client" or "Contract" object.
- *Parent Section* – parent section type (a hierarchy of sections can be created using this field).
- *Parameter* – parameter determining the object type (see Table 2). The parameter code is specified in this field.
- *Action Type Override* – redefines the action type set on the application type level.
- *Mandatory* – indicates whether it is mandatory to fill in this section in the application.
- *Add Data* – additional information. For sections with the "Contract" types, this field may specify the MULTIPLE tag making it possible to use sections to add several contracts on one level of a hierarchy to an application.

Example of an object hierarchy (see Fig. 17):

- Account contract (main object).
  - Card contract.
    - Address



Product Category	Code	Name	Action Type	Object For Type	Is Ready
Issuing	NEW_CARD	New Card for Contract	Add	Contract	Ready

Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
1	Account Contract				Yes		Ready
2	Card Contract	Account Contract			No		Ready
3	Address Information	Card Contract			No		Ready

Fig. 17. Object hierarchy

The Table 2 shows possible parameter values for different type of object.

Table 2. Possible parameter values depending on object type

Object type	Possible parameter values
Address (address type)	Address type code (Full → Configuration Setup → Client Classifiers → Address Types)
Auth Method (authentication method)	Authentication scheme code (OpenWay → Authentication Module → Authentication Types). For this parameter, this value must be specified on the section type configuration level.
Order (standing payment order)	Standing payment order code. For this parameter, the value must be specified on the section type configuration level. The value of the payment type code can be specified (PAYMENT_TYPE=<payment type code>
Usage (usage limiter)	Usage limiter code (Full → Configuration Setup → Products → Service Packs" →[Usage]).
Tariff	The tariff type and tariff code can be specified TARIFF_TYPE=<tariff type>;TARIFF_CODE_EXT=<tariff code>; (OpenWay → Tariffs → Tariff Types & Tariff Domains → Tariff Domains). The tariff role is defined according to the tariff type. If no value is specified for the "Tariff" parameter, the parameter is interpreted as a tariff role.
Limit (credit limit)	Credit limit type: "F" ("Financial" – credit limit), "N" ("NonFinancial" – additional authorization limit).
Instalment	Instalment Scheme code (Instalments → Instalment Configuration → Instalment Schemes, *Service Code* field value).
InstalmentLimit (instalment limit)	Limit type code (Instalments → Instalment Configuration → Instalment Scheme Groups → Instalment Limit Types).
Device Parameters Override (set of device parameters)	Code of the channel for which device parameters will be redefined (Full → Configuration Setup → Main Tables → Message Channels)
Classifier (client or contract classifier)	Classifier code (Full → Configuration Setup → Common Handbooks → User Classifiers). For this parameter, the value must be specified on the section type configuration level.
Parameter (client or contract parameter)	Parameter code (Full → Configuration Setup → Common Handbooks → Contract Parameters Setup).

Object type	Possible parameter values
Contract	Contract category: "M" ("Merchant" – device contract), "A" ("Account" – account contract) and "C" ("Card" – card contract).
Counterparty (payment participant)	Payment type code (Full → Configuration Setup → Transaction Types → Payment on Account Types).
Event	Event type code (Full → Configuration Setup → Products → Issuing Private Products (Issuing Corporate Products) → Issuing Event Types; Full → Configuration Setup → Products → Acquiring Products → Acquiring Event Types).

Rules for Building Schemes

When building an application scheme, the following rules must be observed:

- The type of section for working with a "Client" or "Contract" object can be used one time in the scheme. If it is necessary, for example, to include two sections with the "Contract" type in a scheme (for a merchant contract and a related device contract), two section types for working with the "Contract" object must be created in advance.
- Section types for working with all other object types must be unique within their "branch" of a scheme.
- When working with an application, it is possible to enter information about a number of objects, even if a special section was not created for them. . Forms for working with these objects are available from the form of the parent object – client ("Client Info") or contract ("Contract Info"). A list of these object types if given below:
  - Classifier
  - Credit Limit (for contract only)
  - InstalmentLimit (for contract only)
  - Event (for contract only)
  - Usage (for contract only)
  - Tariff (for contract only)
  - Order (for contract only)
  - Address
  - Affiliation
  - Parameter
  - Contract (if the parent object is an issuing account contract).
  - Card Info (if the parent object is a card contract).
  - Device (if the parent object is an acquiring module contract).

- Sections for working with a "Client" or "Contract" object type can act as the parent to sections of another type. Possible combinations of sections are shown in Table 3.

Table 3. Possible combinations of object types

Section object type	**Possible child section object types**	**Issuing/Acquiring**
Client	AddInfo	Issuing, Acquiring
	Address	Issuing, Acquiring
	Affiliation	Issuing
	Auth Method	Issuing
	Classifier	Issuing, Acquiring
	Contract	Issuing, Acquiring
	Customer Identifier	Issuing
	Counterparty	Issuing
Contract	AddInfo	Issuing, Acquiring
	Address	Issuing, Acquiring
	Affiliation	Issuing, Acquiring
	Auth Method	Issuing
	Classifier	Issuing, Acquiring
	Preferred	Issuing
	Client	Issuing
	Card Info	Issuing, Acquiring
	Contract	Issuing, Acquiring
	Credit Limit	Issuing
	Instalment	Issuing
	Instalment Limit	Issuing
	Customer Identifier	Issuing
	Device Parms	Acquiring
	Device Rec	Acquiring
	Event	Issuing, Acquiring
	Order	Issuing, Acquiring
	Counterparty	Issuing, Acquiring
	Relation	Issuing, Acquiring
	Status	Issuing, Acquiring
	Stop List	Issuing
	Tariff	Issuing, Acquiring
	Usage	Issuing, Acquiring

- It is possible to build a scheme consisting of one section. In this case, the main section can be used to work with any type of object. Here it should be noted that child objects cannot later be added to such a scheme (if the main section is used to work with an object type other than "Client" or "Contract").

Examples of Creating Application Schemes with a Hierarchy of Contracts for Different Clients

WAY4 allows application schemes to be constructed with a multilevel hierarchy of contracts for different clients.

Application schemes for each client (section with the "Client" object type) that is added as a child section must:

- Contain a child section with the "Contract" object type and "Add" action type. An action type can be set in the application type or redefined in a section in the application scheme (*Action Type Override* field).
- Allow an indicator to be set specifying that the section is optional. If the "Client" section is marked as optional ("No" in the *Mandatory* field) and no client identifiers (*Registration #*, *Client Number*, *Social Security #* or *Individual Number*) are specified in the application, this section of the application and all its child sections will be ignored.

Examples of scheme construction are shown below.

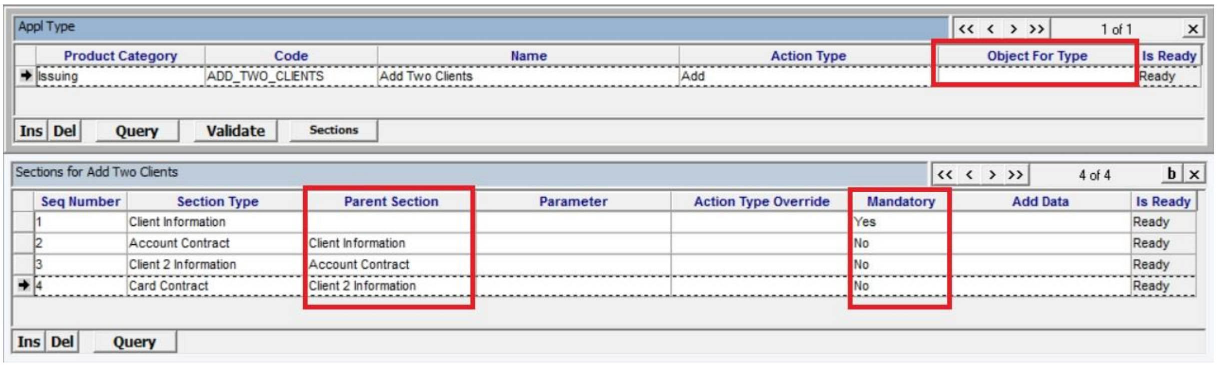
Example 1: Creating a hierarchy of contracts with different clients.

Create a hierarchy of contracts for two different clients: an account contract (client 1) and a subordinate card contract (client 2).

The object hierarchy can be shown as follows:

- Client 1 (main section in the scheme)
  - ▪ Account contract
    - ♦ Client 2
      - Card contract for client 2

Application scheme setup according to this example (see Fig. 18):



Appl Type — 1 of 1

Product Category	Code	Name	Action Type	Object For Type	Is Ready
Issuing	ADD_TWO_CLIENTS	Add Two Clients	Add		Ready

Sections for Add Two Clients — 4 of 4

Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
1	Client Information				Yes		Ready
2	Account Contract	Client Information			No		Ready
3	Client 2 Information	Account Contract			No		Ready
4	Card Contract	Client 2 Information			No		Ready

Fig. 18. Creating a hierarchy of contracts with different clients

The *Object For Type* field to specify the application's main object type must be empty in cases for two different clients, like the one in this example.

To include two sections with the "Client" type in the application scheme, two separate types of section must have been created in advance. In the figure above (see Fig. 18), these are the "Client Information" and "Client 2 Information" sections. For information about creating section types, see the section "Application Section Types".

"Client Information" is the main section in the scheme ("Client" object type). An account contract ("Account Contract" object) is added as a child section for this client. The second client is linked to the first client's account contract. The second client's card contract is a subcontract of the account contract.

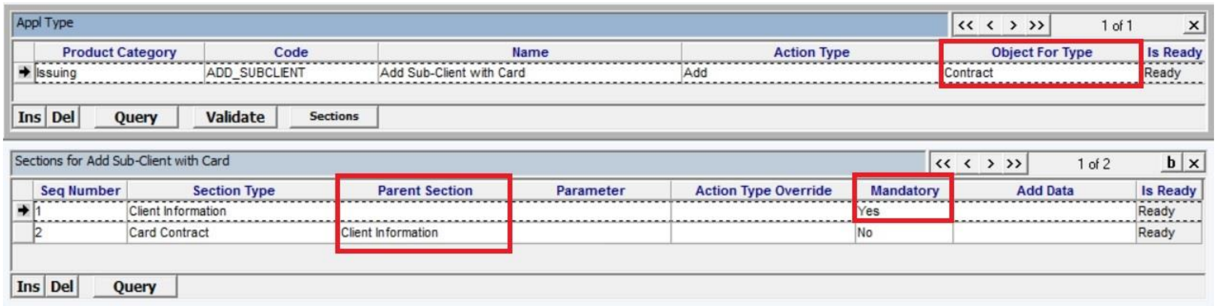
Example 2: Creating a subcontract for a new client who differs from the parent contract's client.

The client (client 1) already has an account contract in WAY4. A new card contract must be created for another client (client 2).

An object hierarchy is created for the account contract in WAY4 and can be shown as follows:

- Client 2 (main section)
    - Card contract for client 2

Application scheme setup according to this example (see Fig. 19):



Appl Type

Product Category	Code	Name	Action Type	Object For Type	Is Ready
Issuing	ADD_SUBCLIENT	Add Sub-Client with Card	Add	Contract	Ready

Sections for Add Sub-Client with Card

Seq Number	Section Type	Parent Section	Parameter	Action Type Override	Mandatory	Add Data	Is Ready
1	Client Information				Yes		Ready
2	Card Contract	Client Information			No		Ready

Fig. 19. Creating a subcontract for a new client who differs from the parent contract's client

The application's main object type is specified in the *Object For Type* field – "Contract". The account contract is the main contract for all contracts created for the new client.

"Client Information" is the main section in the scheme ("Client" object type). A card contract (Card Contract object) is added as a child section for this client.

All contracts for client 2 are subordinate to the account contract specified.