



icIntellect:

Development and Management Guidelines for Data Warehouse and Reporting

eLoyalty Internal Use Only. Confidential and Proprietary.

Version 9.0



IN00-9132-01



System Requirements for Application Installation

eLoyalty Internal Use Only. Confidential and Proprietary.

Version 9.0

IN00-9132-01

First Edition (February, 2014)

Copyright © 2014 eLoyalty Corporation. All rights reserved.

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of eLoyalty Corporation. eLoyalty Corporation makes no warranty of any kind with respect to the completeness or accuracy of this document, or for damages resulting from the use of the information contained herein.

Table of Contents

About This Document	v
Who Should Read This Book	v
Conventions Used in This Book	v
The eLoyalty Library	vi
Chapter 1. Data Warehouse Objective and Goals	1
Chapter 2. Purpose of this Document	3
Chapter 3. Database Objects	5
New Databases	5
Data Type Prefixes	5
Schemas	6
Tables	6
Views	7
Functions	7
Synonyms	8
Stored Procedures	8
Programming Guidelines	10
Security Permissions	11
Chapter 4. Reports	13
Adding Standard Functions to Custom Reports	14
Chapter 5. Database Access	15
Chapter 6. icApplications Report Security Collections	17

Chapter 7. Report Manager	19
Chapter 8. Quality Assurance	21
Chapter 9. Maintenance	23
Chapter 10. Compliance	25
Appendix A. Trademarks	27

About This Document

As the document title suggests, this document describes development and management guidelines for warehouse and reporting. The document covers suggestions and rules for those who develop reports, either standard reports or customer-specific "custom" reports.

The intent is to retain a consistency in how new reports look and feel, either new reports that are intended to become generally available, or new custom-built reports that are developed to meet a specific need for a specific customer.

Who Should Read This Book

This document is intended for anyone who intends to create one or more new reports, either for the icIntellect product or to address a specific customer need for one or more custom-built reports. In addition, the information contained herein also applies to anyone who performs updates to existing reports, either within the icIntellect product or to update a customer-specific report.

Conventions Used in This Book

The following conventions are used in this book:

- Plain text - Window titles, names of toolbar button clusters, and names of tabs. These items are included exactly as they appear in the interface.
- `Monospace` - Names of files, names of directories.
- **Bold** - Column labels, box or field labels, buttons, menu items, commands, note labels, and names of keyboard keys. These items are included exactly as they appear in the interface or on the keyboard.
- *Italics* - Document titles, variables, and emphasis for importance (for example, *Object_Name*).

The eLoyalty Library

This section presents a list of all internal documents that are related to this internal document. This includes documentation for other related products as they come into existence or related documentation that currently exists. This section will be updated periodically to reflect all documentation that is available currently.

INCM-9110-01	<i>icCallMeInstallation Guide</i>	Provides information about installing the icCallMe product.
INIT-9101-01	<i>icIntellect Installation Guide</i>	Provides information about installing the icIntellect product.
IN00-9130-01	<i>System Requirements for Application Installation</i>	Provides system requirements for a variety of eLoyalty products that must be met before a product installation.

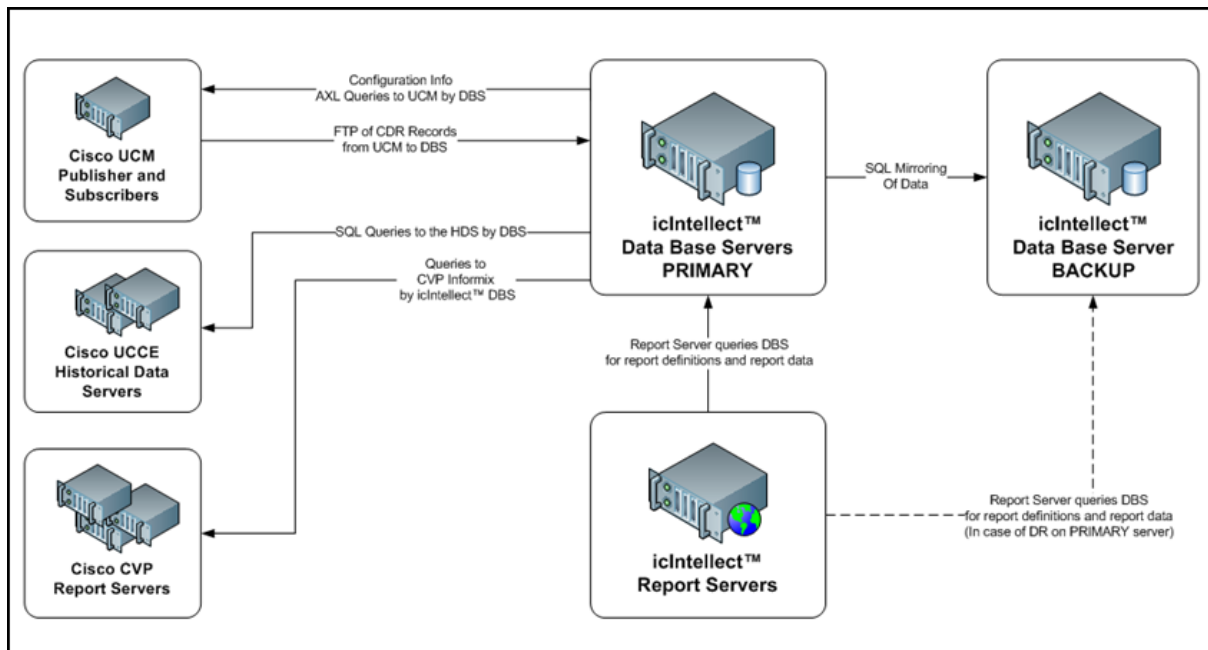
Chapter 1. Data Warehouse Objective and Goals

The fundamental objective of the icIntellect[®] data warehouse is to bring data together for analyzing and summarization for future analysis. Data from various applications and other sources are selectively extracted and organized on the data warehouse database for use for business intelligence reporting.

The development goals are to:

1. Keep reporting data separated from production systems.
2. Provide a single point source for information.
3. Provide easy and secure access to data through Business Intelligence (BI) reporting.

The following diagram provides a high level view of how the data flows for icIntellect data sources.



Chapter 2. Purpose of this Document

The following statement provides guidance for authorized development for the icIntellect data warehouse. No deviation is allowed from the information contained in this document. Contact Product Management for any issues that may arise that are not answered in this document.

The intent of this document is to insure the following:

1. **Performance** – Ensure that any development does not hinder data warehouse performance.
2. **Maintenance** - Database maintenance jobs and SQL Server Agent jobs are not subjective to developers. The processes in place are authorized by Product Management and Product Development and should not be changed without prior authorization. This includes all database schedules for retrieving data.
3. **Proprietary** – eLoyalty® has exclusive legal rights to all icIntellect products. No client is authorized to access any icIntellect database or receive any documentation relating to product design or structure.
4. **Coding Standards** – Ensure that the code is kept clean, understandable, and professional.

Chapter 3. Database Objects

This chapter includes the following sections:

- "New Databases" on page 5.
- "Data Type Prefixes" on page 5.
- "Schemas" on page 6.
- "Tables" on page 6.
- "Views" on page 7.
- "Functions" on page 7.
- "Synonyms" on page 8.
- "Stored Procedures" on page 8.
- "Programming Guidelines" on page 10.
- "Security Permissions" on page 11.

New Databases

New databases require authorization from Product Management. Current databases should be utilized if the data already exists in that location.

If a new database is authorized, it must conform to current standards. The database name should accurately reflect the database content and function. If a new database is required for new icIntellect products, consult Product Management for the naming standard. If a new database is requested and approved for a client, the standards in this section *and* in the following sections will be applied.

For example, if the name of the Client is ABC Corp., the database name would be icIntellect_ABC1.

Data Type Prefixes

The prefixes listed in the following table should be used to describe the object types. These prefixes are placed at the beginning of the object name.

Object Type	Prefix
Tables	ict
Views	icv
Synonyms	ics
Functions	icf
Stored Procedures	icx
Link Servers	ICL

Schemas

All standard database objects will use the "dbo" schema.

All custom database objects will require a different schema based on the client prefix. For example, if there is a custom report for ABC Corp, and a new object is required, the schema would be the client prefix – in this case – "abc." If additional tables are authorized by Product Management the same schema pattern follows. The custom schema should reflect the client code which can be obtained from the icAppsAAA_APM database in the ict_Programs table. The client code is in the ProgramPrefix column.

When creating the custom schema, ensure that the Schema Owner is "dbo."

Tables

Standard and custom reports can utilize all available tables in the dbo schema, however changes to table design or data is not authorized. If the table contains an idProgram field, the logic being developed must be filtered for the specific client.

No additional tables are authorized for creation without prior approval from Product Management and verified by Quality Assurance team. Any required changes require a release from Product Management and verified by Quality Assurance.

Data from tables are not authorized to be exported to any other system without Product Management approval. This includes through views or synonyms.

All system tables will maintain the "dbo" schema. This applies to standard databases only.

No triggers are authorized to be created on any database without approval from Product Management.

No indexes are authorized to be created or altered without approval from Product Management. Naming conventions for indexes are as follows: Only one suffix per index can be appended. The application of the appropriate suffix should follow the following hierarchy: primary key, clustered index, foreign key, other index. For example, an index that is both a primary key and clustered should have a suffix of '_PK'. It is good practice to index columns that are frequently used in a query's selection criteria.

- Primary keys have a suffix of "_PK."
- Foreign keys have a suffix of "_FKx" where x is a number that is incrementally assigned.
- Clustered indexes have a suffix of "_IDX."
- All other indexes have a suffix of "_NDXx" where x is incrementally assigned.

The following standards should be applied:

1. Table column names should not use reserved words or key words.
2. Column names must be consistent throughout the database.
3. Tables with the potential for unlimited growth include a plan for retention before moving to production status.
4. Table columns used in frequent search operations are must be indexed to improve efficiency.

Views

A view is a representation of data. It's mostly used as an abstraction of one or more tables with underlying joins. A view is limited to whatever can be expressed in a single SELECT statement. Custom objects views may be utilized for small dataset returns where no client-specific filtering is required. Views cannot be used as the main data source for a report object.

Custom reports are authorized to use standard views. However, no modifications can be made. If the standard view contains an **idProgram** field, the logic must account for this by filtering for the specific client.

Functions

Table functions are not authorized for use because of poor performance. However, custom scalar functions are allowed for use. Custom functions will utilize the schema naming convention for custom

objects.

Custom reports can utilize standard functions in stored procedures. If the function contains an **idProgram** field, the logic must account for this by filtering for a specific client.

Synonyms

All standard synonyms can be utilized by standard and custom report objects. If the synonym contains an **idProgram** field, the logic must account for this by filtering for the specific client. Custom synonyms utilize the schema naming convention for custom objects.

No custom synonyms are authorized to be created from the icAppsAAA_APM, icAppsUCCE, icAppsETL, Report Server, or ReportServerTempdb database.

Stored Procedures

The following standards should be applied to all stored procedures:

1. The database object used to connect directly to a report is a stored procedure for both standard and custom reports.
2. Stored procedures should be based on the report title. For example, if the report title is "Agents Go Wild" and the client is ABC Corp., then a stored procedure name for a client would be "abc.icx_rptAgentsGoWild". The client code can be obtained from the icAppsAAA_APM database in the ict_Programs table. In the table, the client code is in the **ProgramPrefix** column.
3. Custom reports cannot utilize stored procedures used for standard reports.
4. All database objects or database queries should contain, as a minimum, the following parameters, and the parameters should be utilized in the logic:
 - a. idProgram
 - b. UserID
5. The following items should *not* be used to connect directly to a report:
 - a. Tables (including functions)
 - b. Views. There are some standard items that utilize views and table functions in a need

base only.

- Code must be clean and quotes should be added to completely explain the process.
- Stored procedures must include the following logic for error alert. The logic for the stored procedure must be inserted into the following statement. This ensures that errors will be captured and inserted into the ict_SysLog table that generates an entry into the Windows Event Application Log:

```
USE [Database Name]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [schema].[ProcedureName]
(
    ,@UserID VARCHAR(50)
    ,@idProgram VARCHAR(100)

    --Add additional parameters if needed

)

/*****
Copyright, eLoyalty Incorporated, 2013

Object Name: Procedure Name
Object Type: Stored Procedure

Test: Add test statement

Revisions
Date          Name          Action
*****/

AS

SET NOCOUNT ON

-- BEGIN Logic to capture generated errors.

BEGIN
SET NOCOUNT ON;
BEGIN TRY
```

```
-----  
-- STORED PROCEDURE LOGIC STATEMENT  
-----
```

```
-- END Logic to capture generated errors.
```

```
END TRY
```

```
BEGIN CATCH
```

```
EXEC dbo.icx_SysProcedureCall @ObjectID = @@PROCID;
```

```
END CATCH
```

```
END
```

Programming Guidelines

1. Write comments in your objects whenever something is not very obvious. This helps other programmers understand your code.
2. Do not use column numbers in the ORDER BY clause.
3. Use ANSI-Standard Join clauses.

Example of an unauthorized join:

```
SELECT a.au_id, t.title  
FROM titles t, authors a, titleauthor ta
```

Example of an authorized Join:

```
SELECT a.au_id, t.title  
FROM  
authors AS a  
INNER JOIN titleauthor AS ta ON a.au_id = ta.au_id  
INNER JOIN titles AS t ON ta.title_id = t.title_id
```

4. Ensure that all objects in all query types use aliases. They should be followed by the AS statement. This is required in the SELECT clauses for all field selections.
5. Use the graphical execution plan in Query Analyzer or SHOWPLAN commands to analyze your

queries. Make sure your queries do an "Index seek" instead of an "Index scan" or a "Table scan." Those should be avoided where possible.

6. Do not use SELECT * in queries. Always write the required column names after the SELECT statement.
7. Avoid using functions in the SELECT statement.
8. Avoid using functions in the WHERE clause.
9. Be sure to drop all temp tables at the end of the statement.
10. All tables joins should be utilized WITH (NO LOCK). This also includes synonyms.
11. Every query should start with SET NOCOUNT ON.
12. Use upper camel case for all SQL Key words.
13. Avoid searching using not equal operators as they result in table and index scans.
14. Use WHILE loops instead of CURSOR when needed.
15. When referencing or storing year information use 4 digits instead of 2 digits.

Security Permissions

The SQL Login icIntellectRSUser is utilized for standard and custom database objects created by eLoyalty developers. The naming convention for clients you have authorization to use Report Builder, for example, would be icIntellectRSUser_abc for the client prefix ABC. This is to be applied to all database objects utilized for a custom report in the Permissions tab. The client code can be obtained from the icAppsAAA_APM database in the ict_Programs table. The client code is in the **ProgramPrefix** column.

All logins created, for custom objects created by clients, are only authorized to have db_datareader permissions. Clients are not authorized to have icIntellect login credentials.

Note: Clients are *not* authorized to access any instance of the SQL Server where the icApplications[®] or the icIntellect products are located in a multi-client environment.

Chapter 4. Reports

Because custom reports are client-specific, the design is entirely up to the developer and the client. A separate project file should be created for each client requesting custom reports. These files should be in a company-specific location.

For standard reports there are two templates. These templates contain all standard parameters, standard datasets and formatting. They also contain the standard Tablix, which contains the applicable report items (Links images, Header Statement, Footer Statement, and color standards).

The standard templates are:

- icIntellect Report Template.rdl
- icIntellect Report Template With Groups.rdl

Here are some additional guidelines for developing custom reports:

1. All reports require a Report ID to be shown on the report. These are assigned by the Development team. The custom standard will apply (For client ABC – ic001en_ABC, etc). Internal IDs will still be created once the report is added to Report Manager.
2. Every report requires an idProgram and a UserID parameter –at minimum.
3. Report Properties – the following text fields need to be populated:
 - a. **Author** – eLoyalty.
 - b. **Description** – This should be the code that represents the client for custom reports. The client code can be obtained from the icAppsAAA_APM database in the ict_Programs table. The client code is in the ProgramPrefix column:
 - i. For standard ICM reports: ICM
 - ii. For standard CTG reports: CTG
 - iii. For standard CVP reports: CVP
 - iv. For standard Acqueon: ACQ
 - v. For standard PCS: PCS
 - vi. For standard Store: STORE
 - vii. For client custom reports: Use the client ProgramPrefix
 - viii. For Finesse: FIN
4. Because client-specific reports require a separate project file they should utilize the following

guidelines for client-specific databases only:

- a. Data Sources – These are only requested when the client is creating the report. Standard data sources can be used where applicable. For custom data sources include the client name. For example: "ABC Main"
- b. If a client wants to create a report using data in an existing standard database, a view should be created by Professional Services or Product Development. It should be hard-coded for the specific idProgram and encrypted in the custom database created for the client.
- c. Datasets – Create separate datasets for custom reports that contain the client name. For example: "ABC_Agent_DSP"
- d. Main datasets must contain the letters "DS".
- e. Parameter datasets must contain the letters "DSP."
- f. RDL File – The file name cannot be duplicated from existing file names including stock reports. For example: ABC Agents Go Wild.

Note: It is not necessary to use underscores in report names.

- g. Images – Current standard images can be used in custom reports. New images that are created must follow the same naming convention. For example: "ABC_NewImage"

Adding Standard Functions to Custom Reports

Because custom reports are client-specific, product functions such as Language, Profiles and multiple Time Zones are not required, but can be utilized at the client's request.

Refer to the *icIntellect Design Document* (IN00-9133-01) for implementation guidelines.

Chapter 5. Database Access

Clients are not authorized to access any instance of the SQL Server where the icApplications or the icIntellect products are located in a multi-client environment.

Chapter 6. icApplications Report Security Collections

Access to the icIntellect Report Viewer requires a user login in the icApplications Portal. The user must be assigned to the correct client security group. No individual accounts will be created in Report Manager.

A new report collection will be needed for client custom reports. Those collections will need to be added to the client's specific security group. Custom reports are not authorized to be included with standard report collections. Only one custom report collection is authorized for a client.

Standard report collections for clients are created when a client is added to the Portal. This is a manual process. Refer to the *icIntellect Installation Guide* (INIT-9101-01) for implementation guidelines.

Chapter 7. Report Manager

In order for a report to appear in the icIntellect Report Viewer, two tasks must be completed:

1. For custom reports, the client prefix that is added to the report properties must be added to the `ict_SysReportObjectType` table in the icIntellect database.
2. The job must be set to active in the `ict_SysEtIJob` table located in the icAppsETL database.

Note: For further information see the *icIntellect Installation Guide* (INIT-9101-01) and the *icIntellect Design Document* (IN00-9133-01) .

Chapter 8. Quality Assurance

All database objects and reporting objects will need to go through the same QA process as base product development. A person other than the developer needs to review and sign off on the logic for accountability.

Note: This applies to *all* objects, both standard and custom.

Chapter 9. Maintenance

Any Service Request or Change Request that is submitted needs approval from Product Management, before any solution is applied. There are two reasons for this rule:

- To ensure that any conflicts with releases can be avoided.
- To determine if a solution should actually be a release.

Chapter 10. Compliance

All link servers, data feeds, or any import or export, outside of authorized reporting of data, must be approved by Product Management and Network Engineering.

All SQL Logins created must be approved by Product Management and Network Engineering.

These items require an approved Change Request.

Appendix A. Trademarks

This page lists trademarks and registered trademarks that are included in this document.

The following terms are trademarks or registered trademarks of the eLoyalty Corporation in the United States and/or other countries:

- eLoyalty
- icApplications Portal
- icIntellect

