



Maica Client Management Release Notes V.0.342

Ticket #	Ticket Title	Description
		Major Release: Support at Home Support at Home (SaH) functionality has been introduced into Maica. This release delivers significant updates across both user-facing workflows and administrative processes to support compliance with the new Support at Home framework. What's Changed? This release includes major changes or additions of: • Claiming • Care Recipient synchronisation • Care Agreements and Service Management • Budget tracking and analysis • Bulk synchronisation and data management Further Information

Due to the scale of these updates, detailed guidance is provided in the Knowledge Base. Please use the linked articles below to browse through the details of the changes.

User Guide Articles:

- [Claiming Overview](#)
- [Claiming Process](#)
- [Care Recipient Sync](#)
- [Aged Care Agreements](#) (updated for SaH)
- [Manage Services](#) (updated for SaH with new Budget Trend Analysis and Validation logic)

Admin Guide Articles:

- [Claiming Process](#)
- [Care Budget Management Sync](#)
- [Bulk Synchronisations for SaH](#)
- [Care Recipient Sync](#)
- [Budget Trend Analysis Logic](#)
- [Data Objects > III > Triggers](#)

Additionally, please refer to the guide below to help cover everything you need to check in your Salesforce org and within Maica

- [Get Ready for Support at Home Guide](#)

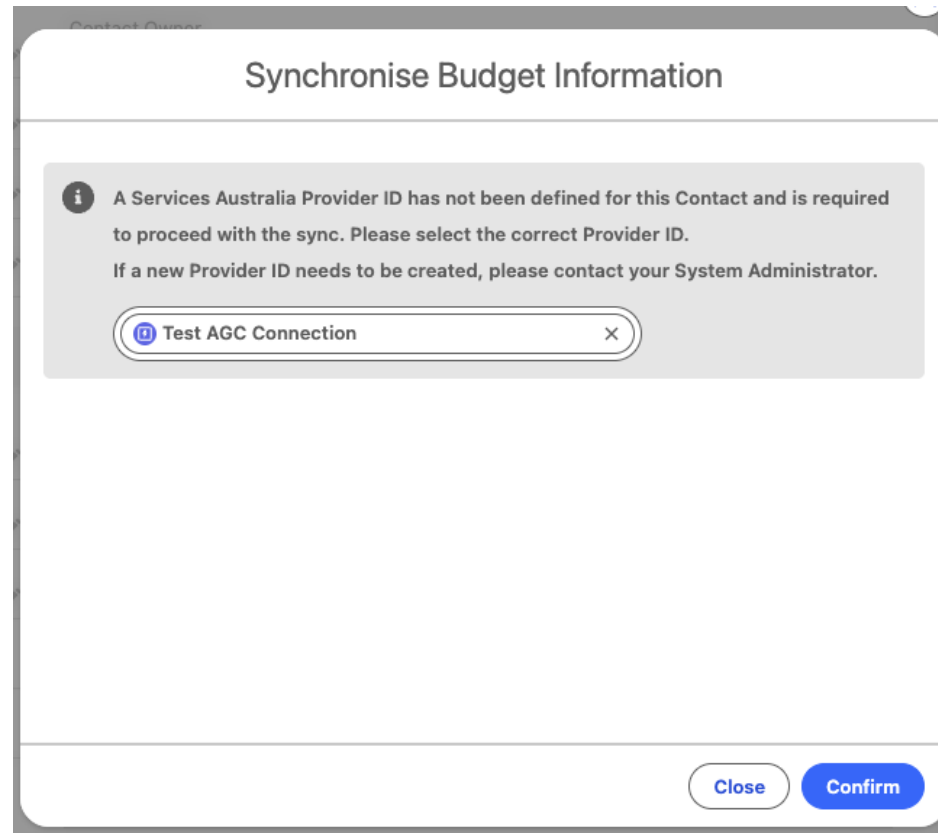
NDIS-950	Support at Home PRODA Integration – Multiple Service Provider ID Support	Enhancement: Multiple Service Provider ID Support Maica now supports management of multiple Service Provider IDs for organisations delivering Support at Home services under different identifiers. This change enables more granular control over which PRODA device credentials are used during service delivery and claim processing, particularly for Participants with multiple Provider IDs registered with Services Australia. What's Changed? • New Provider ID Fields on Contact Object ◦ NDIS_Provider_ID__c: Used for future flexibility in NDIS operations. Typically left empty unless explicitly configured. With a default PRODA device configured, there are no user interface changes as part of the update ◦ Services_Australia_Provider_ID__c: Actively used in Aged Care contexts (Support at Home) to determine which PRODA device to authenticate against. Prompts will be displayed as part of the Budget Sync when this is not set. • User Interface Enhancements ◦ NDIS Users: ■ No visible changes if a default NDIS PRODA device is configured. ■ If no default is set, an error appears during the Synchronise Plan Information quick action (see below). ◦ Aged Care Users (Support at Home): ■ The Synchronise Budget Information quick action now includes a Provider ID prompt when no value is set (see below). ■ If a default exists, it will auto-populate. If multiple IDs are available but no
----------	---	---

		default, user selection is required. ■ Once selected, the Provider ID is stored on the Contact and reused automatically. ● Redesigned PRODA Integration Tab in Client Management Settings ○ The PRODA Integration tab has been fully redesigned to support: ■ Multiple PRODA device registrations ■ Visualisation of device information (including expiry dates) in a new tabular format ■ Device-to-service connection assignment ○ Connection management component featuring: ■ Two tabs: one for NDIS and one for Aged Care (Support at Home) ■ Toggle to set a default API connection ■ Built-in API connectivity test button to verify configuration validity ● Claim Batch and Budget Sync Flow Enhancements ○ Claim batches and sync flows respect the Provider ID assigned to the Contact or Claim Batch. ○ If absent, the system references the default device defined in the Integration tab. ● PRODA Device Configuration and Authentication Logic ○ Flexible routing logic ensures the correct device is used for authentication based on stored IDs or defaults. ○ All PRODA-authenticated operations (Plan Sync, Budget Sync, Claim Submission, Status Checks) now route based on the appropriate stored Provider ID. ● Deprecated Jobs ○ The following scheduled jobs have been removed and are no longer required: ■ PRODA Device Activation Reminder
--	--	---

■ Device Key Expiry Management

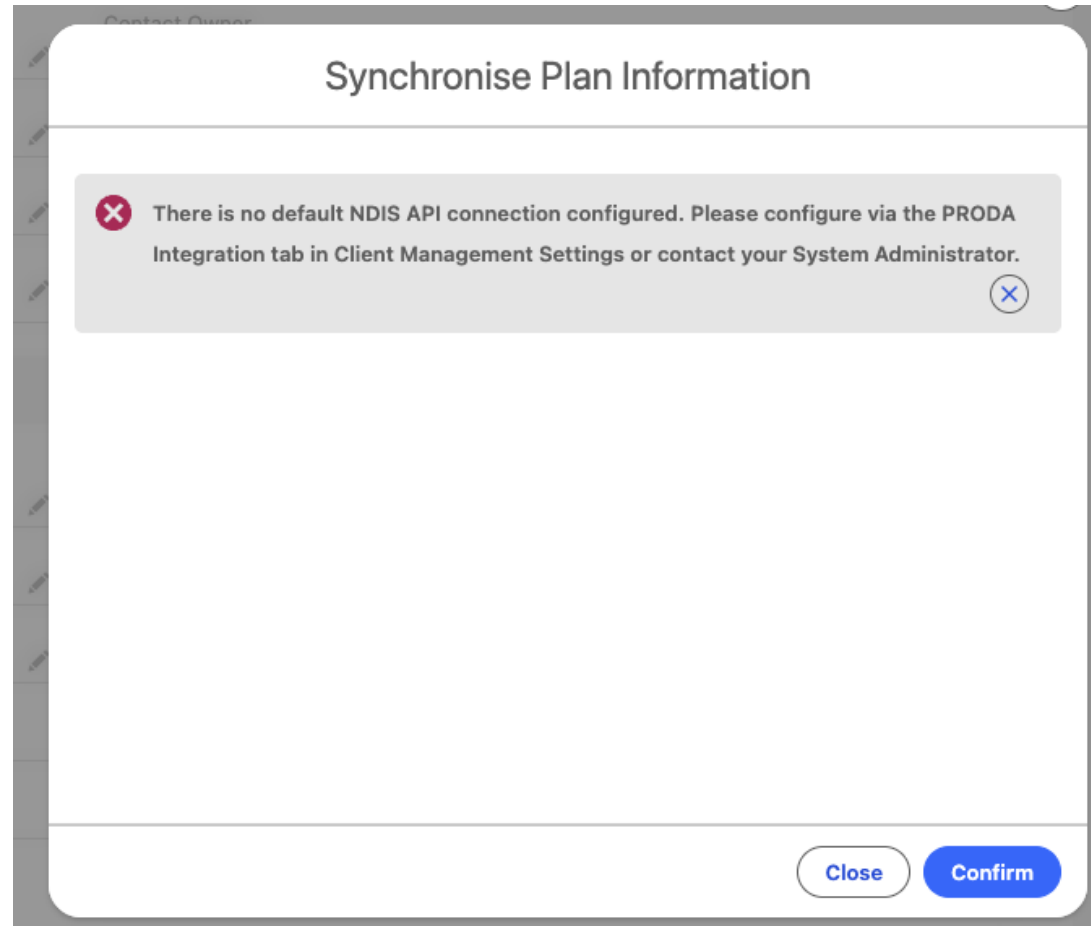
Synchronise Budget Information Quick Action

The image below highlights the updated user interface for the Synchronise Budget Information Quick Action, when the Provider ID selector is displayed.



Synchronise Plan Information Quick Action

The image below highlights the updated user interface for the Synchronise Plan Information Quick Action, when no default NDIS Provider is configured.



Post-Install Steps

Following installation of the Maica v0.342 release, the following steps must be completed to ensure correct operation of the updated PRODA integration:

- Navigate to Client Management Settings > PRODA Integration tab.
 - Verify that all expected PRODA devices are visible in the table at the top of the screen.
 - For each device, ensure there is an associated API connection listed under the NDIS or Aged Care tab (as applicable).
 - If operating with a single Provider ID, toggle the **default** switch to true for the appropriate connection.
 - Use the **Check API Connectivity** button to confirm that each API connection is functioning correctly.
- Open an existing Contact record:
 - Trigger the Synchronise Plan Information quick action (NDIS) to verify the default connection functions correctly.
 - If a Services Australia (Support at Home) connection is configured, also trigger the Synchronise Budget Information quick action to confirm the new Provider ID logic is working as expected.

These steps ensure that Maica is correctly linked to your organisation's PRODA configuration and can authenticate using the appropriate device credentials.

Script to migrate existing PRODA Device Details

Java

```
maica__PRODA_Setting__c device = (maica__PRODA_Setting__c)
maica.vertic_Utils.arrays.firstOrNull([
    SELECT Id
    FROM maica__PRODA_Setting__c
]);

if (device == null)
    device = new maica__PRODA_Setting__c();
if (Type.forName('Schema.' + 'maica__PRODA_NDIS_Setting__mdt') != null ||
Type.forName('Schema.' + 'maica__NDIS_PRODA_Setting__c') != null) {
    Boolean isRecord = Type.forName('Schema.' + 'maica__NDIS_PRODA_Setting__c') !=
null;

    String oldSettingSql = '' +
        'SELECT Id,\n' +
        (isRecord ? 'Name,\n' : 'DeveloperName,\n') +
        'maica__Auth_Client_ID__c,\n' +
        'maica__Client_ID__c,\n' +
        'maica__Device_Name__c,\n' +
        'maica__Device_Expiry__c,\n' +
        'maica__Device_Key_Expiry__c,\n' +
        'maica__Endpoint__c,\n' +
        'maica__Mode__c,\n' +
        'maica__Organisation_ID__c,\n' +
        'maica__Private_Key__c,\n' +
        'maica__Public_Key__c\n' +
        (isRecord ? 'FROM maica__NDIS_PRODA_Setting__c\n' : 'FROM
maica__PRODA_NDIS_Setting__mdt\n') +
        (isRecord ? 'WHERE Name = \'Default\'\n' : 'WHERE DeveloperName =
\'Default\'\n');
}
```

```

        SObject oldSetting = (SObject)
maica.vertic_Utils.arrays.firstOrNull(Database.query(oldSettingSql));

        if (oldSetting != null) {
            device.Name = (String) oldSetting.get('maica__Device_Name__c');
            device.maica__Client_ID__c = (String) oldSetting.get('maica__Client_ID__c');
            device.maica__Device_Name__c = (String)
oldSetting.get('maica__Device_Name__c');
            device.maica__Device_Expiry__c = (Datetime)
oldSetting.get('maica__Device_Expiry__c');
            device.maica__Device_Key_Expiry__c = (Datetime)
oldSetting.get('maica__Device_Key_Expiry__c');
            device.maica__Mode__c = (String) oldSetting.get('maica__Mode__c');
            device.maica__Organisation_ID__c = (String)
oldSetting.get('maica__Organisation_ID__c');
            device.maica__Public_Key__c = (String) oldSetting.get('maica__Public_Key__c');
            device.maica__Private_Key__c = (String)
oldSetting.get('maica__Private_Key__c');
            insert device;
            maica__PRODA_Provider_Setting__c ndisProvider = new
maica__PRODA_Provider_Setting__c(
                Name = 'Default NDIS Provider',
                maica__Service_Type__c = 'NDIS',
                maica__Default_Provider__c = true,
                maica__PRODA_Setting__c = device.Id,
                maica__Auth_Client_ID__c = (String)
oldSetting.get('maica__Auth_Client_ID__c'),
                maica__Endpoint__c = (String) oldSetting.get('maica__Endpoint__c')
            );
            insert ndisProvider;
        }
    }
}

```

		Default Connection Toggle Service Providers API Connections <u>NDIS</u> Aged Care Test  Default NDIS Provider API Connection  * PRODA Device  maica_ndis  Check API Connectivity
NDIS-924	Product - Item Number Override	Enhancement: Product - Item Number Override A new Item Number Override field has been added to the Product object in Maica. This enhancement enables Providers to define their own Support Item Numbers for Support at Home products where variation is required (e.g. pricing based on time of day), addressing

limitations in the standard Services Australia product catalogue.

What's Changed?

- Introduction of **Item Number Override** on the **Product** object.
- Enables service providers to create customised item numbers where:
 - The Support Item definition from Services Australia is not sufficient.
 - No standardised or variable pricing exists.
 - Variants such as daytime, afternoon, evening, and overnight are required.
- Where this field is used:
 - The existing **Support Item Number** field can be left blank.
 - The **Item Number Override** does not need to be unique.
- The Product can then be included in:
 - Price Books
 - Service Agreements
 - Budget Planning and Claims Processes

Please note: It is **not advised** to populate the **Item Number Override** field for NDIS Products and Services.